MIKROSZÁMÍTÓGÉP MAGAZIN

A NEUMANN JÁNOS SZÁMÍTÓGÉP-TUDOMÁNYI TÁRSASÁG LAPJA

1985/4

Tanuláshoz, munkához,
játékhoz pro primo:

PRIMO

személyi számítógép

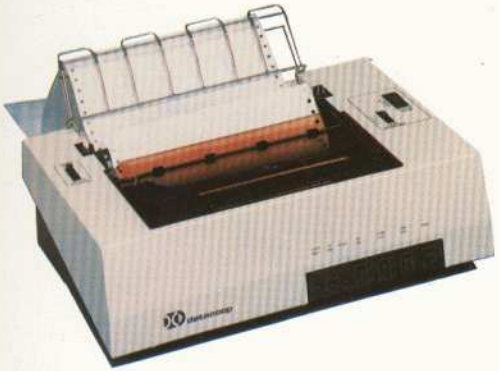
- BASIC programozási nyelv
- billentyűzet: kapacitív elven működő
  magyar ABC kis- és nagybetűkkel
- megjelenítés: 16 sor × 42 betűhely (szöveg)
  256 × 192 képelem (grafikus)
- program és adattárolás, kazettás magnetofonnal
- TV készülékhez csatlakoztatható
- háromféle változat memória kapacitás (kbájt) szerint:

ROM	RAM	Fogy. ár (Ft)
16	16	11 500,–
16	32	16 150,–
16	48	19 339,–

- saját tápegység (4 600,– Ft)

MICROKEY

DCD · PRT · 80
GRAFIKUS MOZAIKNYOMTATÓ



- Nyomtatható karakterek száma soronként: 80
- Nyomtatási sebesség: 80 kar/s
- Karakterkészlet: 95 ASC II/IS
- Kétirányú nyomtatás logikai kereséssel
- Egyidejű másolatok száma: 1 eredeti
  3 másolat
- Mikroprocesszoros vezérlés
- Egykártyás felépítés
- Változtatható papírszélesség
- Grafikus szimbólum készlet
- Bit vezérlésű grafika (tű címzés)
- Traktoros vagy gumigörgős papírtovábbítás
- Papírhiány érzékelés
- Dupla széles karakterek
- Automatikus soremelés
- Festékszalag kazetta
- DZM vagy CENTRONICS interfész

datacoop

Kapható: EMO ELEKTROMODUL

2. sz. bolt
Budapest XIII.
Jászai Mari tér 5.
T: 530-800

MIKROSZÁMÍTÓGÉP MAGAZIN

A Neumann János Számítógéptudományi Társaság lapja

A kiadvány a Tudományszervezési és Informatikai Intézettel együttmûködve készül

A szerkesztő bizottság vezetője:
Kovács Győző

Munkatársak:

Broczkó Péter (hírek)
Buday György István (személyi számítógépek)
Jakab Ágnes (ember–gép kapcsolat)
Kovács Győző (levelezés)
Lindner László (sakkprogramozás)
Pataki Ernő (programozástechnika)
Petróczy Judit (könyvek)
Pogány Csaba (alkalmazástechnika, tanfolyam)
Simonyi Endre (klub)
Takácsy Ildikó (favágás)
Vadkerti János (μprogramok
Varga András (iskola – számítógép)
Vass Nándor (alkalmazások)
Votisky Zsuzsa (játékprogramok)
Zárda Sarolta (piac)

A szerkesztőség munkatársai:

Albert Tibor
Nacsa Sándor

Felelős szerkesztő:
Könyves Tóth Pál
Szerkesztőség:
1027 Budapest II., Fő utca 68.
Telefon: 154-250

Kiadja: a Delta Szaklapkiadó és Műszaki Szolgáltató Leányvállalat
Felelős kiadó:
Faklen Pál igazgató
1442 Budapest, VII., Garay utca 5.
Telefon: 415-583, 215-440

Terjeszti a Magyar Posta
Előfizethető
bármely postahivatalban, a kézbesítőknél, a Posta hírlapüzleteiben és a Posta Központi Hírlap Irodában (Budapest V., József nádor tér 1. Postacím: 1900 Budapest) közvetlenül vagy postautalványon, valamint átutalással a PKHI 215–96162 pénzforgalmi jelzőszámra
Előfizetési díj:
egy évre 180,– Ft,
fél évre 90,– Ft.

Szedte:
a Nyomdaipari Fényszedő Üzem (857711/09)

Nyomás:
Petőfi Nyomda, Kecskemét, Külső Szegedi út 6. (85/50064)
Telefon: 28-777
Felelős vezető:
Ablaka István igazgató

INDEX: 25629
ISSN 0236-6088

Címképünk:
A COMPROJECT GMK COMPUT–80 mikroszámítógépe

Tartalom

A telefonról 2
IBM személyi számítógépek hazánkban 3
A számítástechnika nem játék! 14
A számítógép-hobbizmus úttörői 19
A technológiák szerepe 20
Számítógépesítés és elektronizáció 22
μprogramok 26
Tudáspróba 35
Adok – veszek – cserélek 35

ISKOLA – SZÁMÍTÓGÉP

Egyenletek közelítő megoldása 6
Tisztelt Szerkesztőség! 8
Magyar nyelvű hibaüzenetek 10

TANFOLYAM

Alapozás XI. 12

TERMÉKISMERTETŐ

A Comput–80 mikroszámítógép 16

PROGRAMOZÁSTECHNIKA

LOGO – a programozástanulás új útja 24
Megoldás rekurzív programozással 25

μKLUB

Építsünk számítógépet! IX. 34
Kazettaszabvány 36
Mit érnek a szaklapok? 37
Programmentés magnetofonra 37

JÁTÉKPROGRAMOK

Kalandjátékok 38
Űrhajó 38
Középkor-játék I–II. 39

AZ OLVASÓ ÍRJA 40

HÍREK, ÉRDEKESSÉGEK 43

SAKKPROGRAMOZÁS

Lépésről lépésre 44

KÖNYVEK 45

A telefonról

„Senki sem követ el annyi hibát, mint az, aki csakis megfontoltan cselekszik."
(Vauvenargues)

Tulajdonképpen más témáról akartam a szerkesztőségi cikket írni, de elolvastam a Magyarország 1985/14. számában Antal Anikónak a hazai távközlés problémáiról szóló írását, és ez hozzászólásra ösztönzött.

A leírtakból, amelyekkel nagyon egyetértek, a hazai távközlési hálózat elmaradottságát bemutató néhány adatot ismétlek meg:

- A világon a 41. helyen állunk, ha a távközlésre fordított beruházások és a nemzeti jövedelem viszonyát tekintjük; megelőz bennünket például Törökország, Csehszlovákia, Bulgária, Portugália, Görögország, Ausztria stb.

- A távbeszélőkészülék-sűrűségben sem állunk jobban. Nálunk 6,6 készülék jut 100 lakosra, míg az iparilag fejlett országokban legalább 30, például Franciaországban 42 (a családok 98%-ának van telefonja). A meglevő távbeszélőhelyek elosztása is aránytalan: nagyobbik részük Budapesten működik. Egyes, mezőgazdaságilag fejlett vidéki településeken a helyzet sokkal rosszabb, 1,1 beszélőhely jut 100 főre!

- Magunk is sokszor tapasztalhatjuk, hogy a hazai távközlési rendszer elavult, hiszen a beszélgetések megszakadnak vagy létre sem jönnek, az átvivő kábelek foglaltak, így nem meglepő, hogy a hivatalos statisztika szerint a híváso knak csak 40%-a eredményes.

- Miután nagyon kevés a távhívásba bekapcsolt település, az ország háromnegyed részén az állomások délután 4 és másnap reggel 8 óra között nem hívhatók, és maguk sem kezdeményezhetnek beszélgetést.

- Az országban ma 432 000 telefonigényt tartanak nyilván, a postai beruházási keretekből évente legfeljebb 30 000 új távbeszélőhelyet tudnak létesíteni.

- A meglehetősen későn kialakított adathálózat gyakorlatilag megtelt, pedig nálunk csak 1,2 adatállomás jut 1000 lakosra, huszonötször kevesebb, mint a fejlett ipari országokban.

- Ma kimutathatóan 1 Ft távközlési befektetés 8–10 Ft népgazdasági hasznot hoz. (Szovjet adatok szerint ez az arány náluk 1 : 3,1-hez, míg Franciaországban 1 : 7, 1 : 8-hoz.)

Joggal merül fel a kérdés, hogy Antal Anikó elemző cikke után mi újat mondhat még a μM a távközlési hálózatról azonkívül, hogy egyetértünk: *népgazdasági szinten feltétlenül elsőbbséget kell adni a távközlésnek*, és hozzátesszük: ennek azt kell jelentenie, hogy a távbeszélőhelyek száma rövid idő alatt lényegesen, körülbelül egy nagyságrenddel kell, hogy megnövekedjék.

Azt hiszem, túlzás nélkül megállapíthatjuk, hogy a távközlési hálózat fejlesztésében a fejlett ipari országok mögötti elmaradottságunk nagyjából 10–15 év. A hetvenes évek közepén már Európában is kialakultak az adathálózatok, amelyek az egyszerű felhasználó terminálját a különböző helyekre telepített nagyszámítógépekhez kapcsolták. Hazánkban a számítástechnika széles körű társadalmi elterjesztésének ezt a lehetőségét a szakma kényszerből kihagyta, részben azért, mert nem álltak rendelkezésre megfelelő nagyságú számítógépek, de ha lettek volna is, nem volt olyan távközlési hálózat, amellyel a berendezéseket össze lehetett volna kapcsolni. Nagyon kevés olyan számítóközpont van, amelyet főleg távadatfeldolgozó rendszeren keresztül használnának (100–200 terminál/gép), a 8–10 terminállal üzemelő számítógépeket, amelyek 90%-ban kötegelt feldolgozást végeznek, valójában nem lehet TAF rendszereknek tekinteni. Az is igaz, hogy a terminálok és a számítógépek teljes egészében bérelt vonalakkal vannak összekötve, ami a távközlési hálózat leggazdaságtalanabb felhasználása.

A terminál-számítógép korszak széles körű elterjedésének elmulasztását tulajdonképpen se a népgazdaság, se a Posta nem nagyon vette észre, mert a néhány fejlett kutatóintézet és képzett felhasználó igényét a már említett bérelt vonalas rendszerekkel ki lehetett elégíteni. Ez aztán azt a téves hitet keltette, hogy a meglevő fejlesztési keretekkel a folyamatosan jelentkező új igényeket ha nem is azonnal, de elfogadható időn belül teljesíteni lehet.

Ma már világosan látszik, hogy a népszerűen „mikroelektronikai forradalom"-nak nevezett fejlődésbeli ugrás nemcsak azt jelentette, hogy sohasem hitt mennyiségű, sokat tudó személyi számítógépek jelentek meg a hazai piacon és a felhasználóknál, hanem azt is, hogy igény támadt ezeknek a számítógépeknek az összekapcsolására is. Amíg néhány évvel ezelőtt arról volt szó, hogy egy-kétszáz terminált kell összekötni 10–15 számítógéppel, ma, de legkésőbb „holnap" néhány *százezer* személyi számítógép összekapcsolása lesz a feladat, ezt pedig a jelenlegi távbeszélő-sűrűség és távközlési rendszer mellett lehetetlen megvalósítani.

Van még egy nagyon lényeges különbség a néhány év előtti helyzethez képest. Akkor a távkapcsolat igénye a közületi szférában jelentkezett, ami távközlési szempontból mindig jobban állt, mint a magánszektor. Ma a helyzet fordított. A számítógépek többsége (természetesen a számítógép-kapacitás kisebbik része) magánkézben van – igaz, ezeknek csak töredéke alkalmas még távkapcsolatokra. Nem nehéz megjósolni, hogy már nem sokáig, hiszen a mai Sinclairek, VC20-asok, de a HT–2080 és a Primo is főleg a serdülő ifjúság tanulógépének számít. A komolyabb munkák már a középiskolás környezetben is a Commodore 64, a felsőoktatásban és az intézményeknél pedig az M08X, a Janus, a TAP 34, a Syster és a Proper, illetve az IBM PC, Apple gépeken futnak, amelyeket hálózati rendszerekben is lehet üzemeltetni. Valószínűleg lesznek, akik vitatkoznak velem, de én azt hiszem, hogy nemsokára az utóbbi gépek lesznek túlsúlyban nemcsak az intézményeknél, de az iskolákban és a magánszemélyeknél is, ami a jelenlegi távközlési igényt felfokozza és társadalmi szintre emeli. A telefon munkaeszközzé válik, és olyan fontos lesz, mint a villany, a víz, a csatorna és a többi közüzemi szolgáltatás, ezért minden lakáshoz a távkapcsolatot is meg kell teremteni.

A Magyarország cikke úgy becsüli, hogy a jelenleg tervezett lassú ütemű fejlesztés mellett 1980–2000 között a távközlés elmaradottságából származó népgazdasági veszteség elérheti a 150–250 milliárd forintot. A Posta szerint kb. háromszoros hálózatbővítés (20 főállomás/100 fő) mai árakon mintegy 90 milliárd forint beruházást igényel – és ha az a becslés is igaz, hogy egy távközlésre fordított forint 8–10 forint éves népgazdasági hasznot jelent, akkor az elkövetkező 15 évben fokozott ütemű, folyamatos beruházást feltételezve, az üzleten a népgazdaság 8–900 milliárd forintot nyer.

A beruházásnak a részben pénzben kifejezhető, részben nehezen forintosítható hasznát teljes egészében felsorolni biztosan nem lehet, ebből csak néhány példát szeretnék kiemelni.

A következő években valószínűleg nagyon sok otthoni munkahely fog kialakulni, hiszen a programfejlesztési, adatbeviteli, bizonyos irányítási munkákhoz a hálózatra kapcsolt házi számítógépet lehet majd használni. Ez persze azt jelenti, hogy aki otthon dolgozik, annak nem kell irodaház, íróasztal, portás, üzemi étkezde, de nem lesz szüksége bölcsődére, a gyerekének nem kell napközibe mennie, nem használja reggel és délután a tömegközlekedési eszközöket, nem használja az autóját és ezért nem fogyasztja a drága benzint, de nem szennyezi a levegőt sem. Kevesebbet jár a közhivatalokba, hiszen ügyeit elintézheti akár írásban is a számítógépével. Nem jár a boltokba, ha a rendelését felveszi a raktáráruház. Megvalósítható lesz az elektronikus pénz, mert késedelemmentes távkapcsolat építhető ki a legeldugottabb kifizetőhelyek és a központi adattár között. Talán nem lesz irreális a hitelkártya-rendszer létrehozása sem, amivel mindenki, az állam is és az állampolgár is csak jól jár, hiszen kevesebb pénz lesz forgalomban, a kifizetéseket gyakorlatilag pénzforgalom nélkül lehet intézni.

A fejlett távközlési hálózatnak lesz pénzben ki nem fejezhető hatása is. Sokszor, sokféle fórumon elhangzott már, hogy az értelmiségnek – még az agrárértelmiségnek is – a főváros, illetve a nagyobb vidéki városok felé való mozgása fokozódik. Nem hiszem, hogy ennek az az oka, hogy a tanyaközpontban vagy a falvakban nincs színház vagy mozi. Van televízió, és színház-, moziigényét havonta egyszer-kétszer bárki különösebb anyagi megterhelés nélkül a legközelebbi városban ki tudja elégíteni.

Az alkotó értelmiségnek sokkal inkább a konstruktív munkához szükséges közeg hiányzik. A napi kapcsolat a kollégákkal, a tudományos-szakmai központokkal, szüksége van arra is, hogy az új eredményekről, a legújabb közleményekről, könyvekről napi tájékoztatást kapjon, hogy saját eredményeit bemutathassa és megvitathassa, hogy bármikor kérdezhessen és kérdezzék. Nem beszélve a nemzetközi kapcsolatokról, amelyeket ma, az egész földgolyóra kiterjedő adathálózatok korában a világ bármelyik csücskével ki lehet építeni. A példákat naphosszat lehetne sorolni, de talán ennyi is elég.

Szívesen emlegetik, hogy az informatika korába léptünk, de ehhez az informatika korának eszközei is kellenek: számítógépek, programok, távkapcsolat. Az informatikát ma egyszerre tanulja felnőtt és gyerek, öreg és fiatal, ezért társadalmi méretű oktatási programra van szükség, a megfelelő eszközbázissal együtt.

Az első lépést az iskolaszámítógép-programmal, a számítástechnikai távoktatással, az egész országra kiterjedő μklub-hálózattal már megtettük. Ezek olyan eredmények, amelyekből csak fejlett távközlési rendszerrel lehet továbblépni. Nem kell jóstehetségnek lenni, hogy tisztában legyünk azzal, ha ez a továbblépés rövid idő alatt nem történik meg, akkor nemcsak a létkérdésnek számító elektronizálási program eredményeit, de a jövő szempontjából legalább annyira fontos társadalmi program sikerét is kockáztatjuk.

KOVÁCS GYŐZŐ

IBM személyi számítógépek hazánkban

Az idén április óta nálunk is megrendelhető IBM személyi számítógépekből (lásd a táblázatot) ma már közel 4 millió van alkalmazásban szerte a világon. Egy ilyen horderejű gyártmányról meglehetősen visszás dolog lenne termékismertető értékelést írnunk, már csak azért is, mert azok, akik esetleg kedvet kapnának a vásárláshoz, nem biztos, hogy egykönnyen elő tudnák teremteni a szükséges konvertibilis valutát, és a vétel még az Egyesült Államok szállítási engedélyt adó hatóságain is múlik. A szerencsés keveseknek elég az elinduláshoz szükséges alapvető tájékozottság, ami nem árt a kívülrekesztett többségnek sem, hogy legalább sejtése legyen az irodai és professzionális számítógép-alkalmazások jelenét az egész világon tömegesen meghatározó technikáról.

Alapmodellek	Megjelenítés/nyomtatás változat					
	Mo G	C G	E G	Mo Q	C Q	E Q
PC 256 k, 1 × 2 320 k	4 000	4 446	–	5 481	6 127	–
PC XT 256 k, 1 × 2 360 k	4 394	4 840	–	5 875	6 521	–
PC XT 512 k, 1 × 2 360 k		5 331	5 973		7 012	7 654
PC XT 256 k, 1 × 360 k W 10 M	5 866	6 312	–	7 347	7 993	–
PC XT 512 k, 1 × 360 k W 10 M	6 357	6 803	–	7 838	8 484	–
PC AT 512 k, 1 × 1,2 M 1 × 360 k W 20 M		8 891	9 533	–	10 572	11 214
PC AT 1 M, 1 × 1,2 M 1 × 360 k W 20 M		10 263	10 905	–	11 944	12 586

Jelmagyarázat: k = kilobájt, M = megabájt, f = floppy-meghajtó, W = Winchester-meghajtó, Mo = monokromatikus megjelenítő, C = színes megjelenítő, E = növelt képességű színes megjelenítő, C = Quietwriter nyomtató, G = grafikus nyomtató

Az IBM Magyarországi Kft által bejelentett IBM személyi számítógép-konfigurációk jellemzői és árai US dollárban

Nagy teljesítményű alapgépek

A PC és PC XT modellek Intel 8088 típusú mikroprocesszora 8 bites adatsínt használ, de műveleteiben 16 bites. Órafrekvenciája 4,77 MHz. A maximális teljesítményt 250 ezer művelet/mp-ben határozza meg az IBM.

A PC AT modell az Intel legújabb mikroprocesszorán, a 80286-on alapszik. Ez a szupergyors processzor 16 bit szélességben forgalmazza az adatokat a memóriabuszon, belső megvalósítása is hatékonyabb, órafrekvenciája pedig 6 MHz. Az Intel mérései szerint legalább háromszor gyorsabb a korábbi processzornál, sőt bizonyos feladatoknál eléri a nyolcszoros sebességet, amivel már a VAX–11/780 szupermini teljesítményéhez közelít; hasonló processzor-teljesítmény, mint az IBM 4300-as sorozat felsőbb tagjainál.

Külön előny, hogy a nagy pontosságot igénylő, számításigényes alkalmazások numerikus műveleteit külön matematikai társprocesszor chip (Intel 8087, ill. 80287) támogatja, amit az alapkártyán kialakított foglalatba kell bedugni. Az egyébként szoftverben megvalósítandó műveletek sebessége ezzel akár két nagyságrenddel növelhető: a 32 bites és 64 bites lebegőpontos szorzásé például a 80-szorosára (a műveletvégzési idő kb. 24 μs).

A processzor-funkciókat megvalósító alapkártyán max. 256 kbájt (PC és PC XT), ill. 512 kbájt (PC AT) kapacitású felhasználói memória és 5 db (PC), ill. 8 db (PC XT és PC AT) kártyacsatlakoztatási hely is található. Az utóbbiak felhasználásával lehet a tárat bővíteni 640 kbájtra (PC és PC XT), ill. 3 Mbájtra (PC AT), valamint illeszteni a gép használatához szükséges perifériákat a megfelelő adapterkártyák segítségével. Egy külön bővítőegység segítségével még további 6 db kártyahellyel növelhető a PC és a PC XT modularitása.

A rendszer erőforrásaival, így a processzorral, a memóriával, a háttértárolókkal és más perifériális egységekkel a DOS-nak nevezett, lemezes operációs rendszer gazdálkodik. Maximálisan 640 kbájt tárat tud kezelni. Az AT fennmaradó tárterületét más módon, ún. virtuális lemezek formájában szolgáltatja. Az eredeti DOS 1.0 változat 160 és 320 kbájtos hajlékonylemez-formátumai mellé a DOS 2.0 bevezette a 180 és 360 kbájtos formátumokat és az XT 10 Mbájtos Winchestereinek kezelését. A PC AT-hez kidolgozott 3.0 változat az új típusú, 1,2 Mbájtos floppy-meghajtók és a szintén új, 20 Mbájt kapacitású Winchester-meghajtók bevezetését tette lehetővé. Típusonként maximálisan 2 db meghajtót kezel a rendszer.

A PC AT új meghajtói nemcsak nagyobb kapacitásúak, hanem sokkal gyorsabbak is. A korábbi Winchester 90 ms-os átlagos elérési idejével szemben 40 ms-ra csökkent az elérési idő, és még az új floppy-meghajtóknál is mindössze 94 ms (a korábbi hajlékonylemezeknél ennek többszöröse). Az elérést követő információátvitel maximális sebessége a Winchesterekénél 5 Mbit/s, az új floppyknál 500 kbit/s, míg a korábbiaknál 250 kbit/s. Ezek az alapparaméterek jól érzékeltetik az egyes alapgépek között lemezes állománykezelési szempontból fennálló teljesítménykülönbségeket, a pontos viszonyt természetesen csak megfelelő próbafuttatásokkal lehet megállapítani.

Az új DOS változatok új szolgáltatásokat is nyújtanak a régebbihez képest. A korábban említett virtuális lemez például a 3.0-ban jelent meg. Az újabb DOS változatok a régebbi gépeken is használhatók, a felhasználónak viszont számítania kell a megnövekedett memória-helyfoglalásra.

Az IBM gondosan ügyel a felülről való kompatibilitásra, így a korábbi változatokra írt programok túlnyomó többsége futni fog az új változaton is. Az 1,2 Mbájtos meghajtóknál a cég megvalósította a korábbi 160/180 és 320/360 kbájtos hajlékonylemez-formátumok kezelésének üzemmódját is. A régi meghajtókon írt lemezeket így minden további nélkül el lehet olvasni az új AT meghajtó segítségével is, de a fordított irányú adathordozó-kompatibilitást nem sikerült megvalósítani. A nálunk bejelentett AT konfigurációkba nyilván ezért van egy régi típusú meghajtó is beépítve.

A PC AT esetében egy tömegesen forgalmazott, új operációs rendszerrel is megjelent az IBM. A Unix System III-mal kompatibilis IBM Personal Computer Xenix többfelhasználós és felhasználónként akár több feladatot is futtatni képes, nagy teljesítményű operációs rendszer. Támogatja a PC AT teljes erőforrásrendszerét, így a maximálisan 3 Mbájt memóriát és a hozzá való 80287-es társprocesszort is. Szolgáltatási rendszere is jóval kiterjedtebb, mint a DOS-é, ami alkalmassá teszi az AT-t arra, hogy általános célú gépként használják mérnöki, tudományos, szoftverfejlesztési, kiadványszerkesztési és sok más területen. Az IBM max. 3 felhasználó kiszolgálására ajánlja a Xenixet.

Képernyős megjelenítési lehetőségek

Ha kizárólag szöveges információt kell megjeleníteni, akkor a *monokromatikus megjelenítő* változat a legelőnyösebb. Egyenként 80 karakterből álló, összesen 25 sornyi információ látható így egyszerre a képernyőn. 9 × 14 képpont méretű mező 7 × 9 méretű pontmátrixában képződnek a karakterek, és a lelógó szárú betűk is a megszokott alakban jelennek meg a képernyőn. Egyes karakterek nagyobb fényerővel vagy inverz alakban megjelenítve kiemelhetők. Lehetőség van az aláhúzásra és villogtatásra is. A képernyő tükrözésmentes, a megjelenítés színe szemet kímélő zöld.

Szövegek színes kiemelésére, irodai/üzleti jellegű grafika ábrázolására a *színes megjelenítő* változat a leggazdaságosabb megoldás. Szöveges üzemmódban 80 vagy 40 karakterből álló, 25 sornyi infor-

máció jeleníthető meg egyszerre. A karakterek kisebb felbontásban (7 × 7 méretű pontmátrix) és szorosabb elrendezésben (8 × 8 képpontnyi karaktermező) jelennek meg a képernyőn, mint az előbbi változat esetén. Előnyös viszont, hogy a max. 8 háttérszín bármelyikében max. 16 előtérszín alkalmazható.

A grafikai üzemmód nagy felbontású, 640 × 200 pontos változata *csak* fekete-fehér megjelenítést támogat. Színes grafika közepes, 320 × 200 képpontos felbontásban áll rendelkezésre. Ekkor két színkészlet egyikét lehet használni. Minden készletben három előtérszín és egy háttérszín van.

Szövegek sok színnel való megjelenítésére és irodai/üzleti célú grafika kontrasztos, nagy felbontású és többszínű ábrázolására legalkalmasabb a *növelt képességű színes megjelenítő*. A monokromatikus és színes változatok emulálása mellett a színes grafikai üzemmódban 640 × 350 képpont felbontású a képernyő, 4 színt lehet választani egy 64 színből álló palettáról, és egy külön megvásárolható bővítőkártya segítségével az egyidőben használható színek száma 16-ra növelhető.

Ez a megjelenítési változat így alkalmas műszaki-tudományos célú, színes grafikai alkalmazásokra is, jóllehet a teljes IBM kínálatban egy még nagyobb felbontású változat (640 × 480, 256 szín 4096 színből álló palettáról) a legmegfelelőbb erre (IBM PC Professional Graphics Display).

A 64 színt tartalmazó palettáról 16 színt lehet használni szövegek igen jó minőségű (8 × 14 méretű mező karakterenként), színes megjelenítésére. Speciális lehetőség a szövegek 43 soros kiíratása (8 × 8 méretű mezőben).

Nyomtatási változatok

Nagyobb mennyiségű, viszonylag gyakori nyomtatási igények esetén, amikor általában a levél minőségű iráskép sem kifejezett követelmény, mindenképpen a *grafikus nyomtató* változatot célszerű használni. Ez a nyomtató megfelelő üzemmódban a képernyőn megjelenített grafikát is képes kinyomtatni (ún. bitképnyomtatás).

A 9 × 9 méretű pontmátrixban mindkét mozgási irányban képezett karaktereket max. 80 karakter/mp sebességgel írja ki a berendezés. Egy sorban maximum 80, 132, 40 vagy 66 karakter lehet, annak megfelelően, hogy a kiírás normál, sűrített, széthúzott vagy széthúzott és sűrített módban történik. A dupla nyomtatás, alsó index, felső index, aláhúzás és felső index lehetőségei révén összesen 18-féle szövegnyomtatási jelleg áll a felhasználó rendelkezésére.

Szélperforált, végtelenített számítástechnikai papírt (ún. leporellót) kell használni, amelynek szélessége 4 és 10 hüvelyk (101,6 és 254 mm) között bármilyen lehet. A lapmagasság beállítható, a papír végét hangjelzés adja tudtunkra. Az eredeti mellett két másolatot lehet egyidejűleg nyomtatni.

Kifejezetten igényes, levél minőségű nyomtatás céljaira a *Quietwriter* fantázianevű nyomtató alkalmazása ajánlott. Az előző nyomtatótól eltérően ez nem ütéses alapelven működik, hanem egy olyan egyedülálló, IBM hőnyomtatási technológián, amely közönséges papíron, speciális festékszalagon keresztül, 40 pont magasságban képezi a karaktereket. Az írásjeleket hüvelykenként (25,4 mm) 10, 12 és 15 karakternyi sűrűségben lehet kiírni, a sebesség ennek megfelelően 40, 48 és 60 karakter/mp. Arányos szélességű proporcionális nyomtatási lehetőség is van (gondoljunk itt az i és m betű szélességének különbözőségére). Négyféle írásjeltípus áll rendelkezésre (Courier 10, Prestige Elite 12, Prestige 15 és Boldface), amelyek közül egyszerre kettőt lehet használni. A nagy felbontású pontmátrixon alapuló hőnyomtatási elv miatt az írásjeltípusok programja ugyanis változtatható, és az Electronic Font elnevezésű, kisméretű ROM kazettáknak két hely van a gépben. Az alkalmazott papír szélessége kézi adagolás esetén 76 és 381 mm között bármilyen lehet. Automatikus lapadagolásnál a legkisebb szélesség 210 mm. Az elérhető maximális nyomtatási szélesség mindkét esetben 333 mm.

Széles körű szoftverellátás

Ma már szinte nyilvántartásba sem lehet venni azt a hatalmas mennyiségű szoftvert, amely az IBM személyi számítógépekre rendelkezésre áll szerte a világon. Az IBM közel 200 szoftvertermékét hirdet katalógusában. Ennek is csak kis hányada az IBM által kidolgozott és folyamatosan támogatott termék, bár éppen ezeket árazzák a legagresszívebben.

Az egyes termékek ára egyébként meglehetősen széles tartományban mozog. Az egyszerű segédprogramokat már 50 US dollárért is meg lehet venni, míg egy BSC vagy SNA adatátviteli emulációs szoftver majd 900 dollárba kerül. A tipikus ár azonban 150 és 600 dollár között változik: a külön kapható szövegfeldolgozó, állománykezelő, tablózó, üzleti grafika készítő és számtáblázat-kezelő programok vannak az alsó határ közelében, az ilyen funkciókat integráltan támogató szoftverek pedig a felső határ környékén. A fordítóprogramok általában 300 dollár körüli árért vásárolhatók meg, noha a DOS COBOL fordító ára 800 dollár.

A szoftverválaszték és az alkalmazási területek közötti összefüggést korábban már részletesen elemeztük (lásd 1985/2. szám, 24–25. oldal). Az ott említetteken túlmenően ki kell emelnünk, hogy az IBM személyi számítógépek nagy teljesítménye számtalan laboratóriumi, ipari, speciális oktatási és sok más alkalmazásban lehetővé tette, hogy a PC és a PC XT a korábban szinte kizárólag alkal-

Az IBM személyi számítógépek családjának legújabb és legnagyobb teljesítményű tagja, az IBM PC AT

Az IBM PC XT modell belülről. Bal oldalon látható az alapkártya vége és néhány behelyezett bővítőkártya. Középen helyezkedik el a floppy meghajtó, jobb szélen a Winchester lemez egység, hátul pedig a tápegység.

mazott miniszámítógépek versenytársa legyen. A PC AT ez irányban hozhat jelentős változást, hiszen képességei már jóval meghaladják egy jelenlegi személyi számítógép-alkalmazás teljesítményigényeit.

A PC Xenix Software Development System egy komplett C nyelvi fejlesztő környezet, keresztfejlesztési lehetőséggel a DOS rendszer irányában. A PC Xenix Text Formatting System a személyi számítógépes dokumentumfeldolgozást minőségileg felülmúló szövegfeldolgozási környezetet ad. A végleges formátumú dokumentumot nemcsak a szokásos számítástechnikai nyomtatókon, hanem fényszedőgépen is ki lehet „iratni". A Xenixhez fellelhető kész szoftvertámogatás persze még csak gyerekcipőben jár a DOS-éhoz képest.

A DOS alapú gépek továbbra is intenzíven bővülő szoftver- és alkalmazási világának a jövőben kulcsfontosságú területe a műszaki grafika. Az IBM szándékokat világosan mutatja, hogy a növelt képességű színes megjelenítő és a még annál is nagyobb grafikai teljesítőképességű másik rendszer (Professional Graphics Display) megjelenésével egyidőben két fontos grafikai alapszoftvert is piacra hozott a cég. A Graphics Development Toolkit a megjelenítő típusától független szoftver csatlakozási felületet nyújt, míg az IBM PC Engineering/Scientific Series részeként megjelent PC Graphics Kernel System (GKS) egy igen hatékony, magas szintű grafikai nyelvet valósít meg, amely még könnyebbé teszi a mérnöki/tudományos alkalmazások fejlesztését. Sajnos ezeket a szoftvereket nem jelentették be nálunk.

Hazai szempontok

Az IBM személyi számítógépek nagy teljesítményű változataival komoly forradalmat *lehetne* elindítani a hazai számítógépesítésben. Ehhez először is a jelenleginél lényegesen több devizára lenne szükség. Másodszor: előbb pozitív tapasztalatokat kell szereznünk a nagyobb teljesítményű konfigurációk exportengedélyezését illetően. Harmadszor: megfelelő felkészültségre lenne szükség a tényleges hazai igényeknek leginkább megfelelő konfigurációk összeállításában.

A kínálat előterében álló típuskonfigurációk (l. táblázat) a tipikusan nyugati szemléletű személyi számítógép alkalmazásoknak felelnek meg. Néhány ilyen példa:

- sok adaton alapuló, maximális gépi kiszolgálással megvalósított (színes grafika), igen magas megjelenítési szintű, komplex döntéstámogatási rendszer;
- professzionális ügyintéző nagy termelékenységű munkavégzését támogató, személyhez igazított nyilvántartási és feldolgozási rendszer stb.

Az ilyen alkalmazások nálunk sem kizártak. Ugyanakkor valószínűbbnek tűnik, hogy a szervezetorientált adat- és információfeldolgozás kerül majd az előtérbe. Ehhez pedig egészen más konfigurációra lenne szükség. (A piaci tapasztalatok és a hazai igények figyelembevételével az IBM kész változtatni a típuskonfigurációkon. A szerk. megj.)

Jó példa a nyomtatók esete. A Quietwriterrel ellátott konfigurációk nem alkalmasak egy szervezet (pl. egy tervező iroda) dokumentumainak rendszeres nyomtatására. A speciális festékszalagot kb. 100 gépelt oldalnyi kiírást követően ki kell cserélni. A fejek cseréje kb. 2500 gépelt oldalnyi információ nyomtatása után válik szükségessé. A grafikus nyomtató ennél lényegesen többet bír, de ezt sem a számítóközpont jellegű, folyamatos listázásra fejlesztették ki.

A szerintünk jobban propagálandó, *elemszintű IBM kínálatban* 216 hardver, szoftver és dokumentációs tétel között lehet válogatni. Megfelelő teljesítményű és kiépítettségű adat- és információfeldolgozó gépet minden további nélkül konfigurálni lehet, hiszen a vevő így máshonnan is beszerezhet elemeket. Jó példa lehet az ékezetes karaktereket „ismerő" terminálokkal és megfelelő nyomtatókkal kialakított, többfelhasználós Xenix rendszer.

A nagyobb teljesítményű általános konfigurálhatóság szempontjából nem egyértelmű a két helyi hálózati rendszer (PC Cluster és PC Network) jelenlegi elérhetősége. A nagy teljesítményű, interaktív gépi grafikán alapuló műszakitudományos alkalmazásokat is korlátozza, hogy az Engineering/Scientific Series elemei közül, egyelőre, csak a professzionális grafikus megjelenítőt kínálják, ugyanakkor hiányzik a más országokban már kapható fejlesztő és támogató szoftver. A meglévő központi adatfeldolgozó rendszerek intelligens termináljaként való csatlakoztatáshoz viszont teljes körű hardver- és szoftverválaszték áll rendelkezésre.

A fenti kínálati korlátozások ellenére úgy véljük, hogy az irodai, részleg, sőt középvállalat szintű szervezetek többségének gyors számítógépesítése jól *megalapozható* lenne a kínált IBM gépekkel. Itt különösen a többfelhasználóssá is kiépíthető PC AT-re gondolunk. Az árak tekintetében is feltétlenül üdvözölnünk kell az IBM gépek megjelenését a hazai piacon. A főbb piacokon alkalmazottaknál mindössze 10–25%-kal magasabb IBM árak mindenki számára érzékelhetővé teszik a belföldi árszínvonal túlzottan magas voltát. A nyugati vevőhöz a hivatalos IBM áraknál általában 10%-ot meghaladó kereskedelmi árkedvezménnyel csökkentett tényleges áron jutnak el a gépek.

A FIXPONT-ITERÁCIÓ – HT ISKOLASZÁMÍTÓGÉPEN

Egyenletek közelítő megoldása

(1) COS(X) − X = 0

Feladatunk az, hogy megkeressük azt a valós számot, amely egyenlő a saját koszinuszával. Ha ezt a problémát számítógép segítségével szeretnénk megoldani, akkor többfajta megoldási formát is ismerünk, például az intervallum-felezéses módszert. Most egy olyan eljárást mutatunk be, amely a legegyszerűbben programozható, de a középiskolai oktatásban ennek ellenére kevesen használják. Az eljárás lényege a következő.

Az (1) egyenlettel ekvivalens az alábbi:
$X = COS(X)$

Legyen X_0 tetszőleges valós szám. Az X_1 legyen $COS(X_0)$, az X_2 legyen $COS(X_1)$, és általában $X_n = COS(X_{n-1})$ (n pozitív egész szám). Ezzel definiáltunk egy (X_n) sorozatot.

Készítsük most el a következő programot, amelynek folyamatábrája az *1. ábrán* látható:

```
 5 REM FIXPONT - ITERACIO
10 CLS:J=0 : N=0
20 INPUT"A kezdoertek ";X
   : PRINT@40,"Tovabb   -T";
     PRINT@104,"Uj ertek -U"
30 N=N+1 :X=COS(X)
40 J=J+1 : IF J=14 THEN J=1
50 PRINT@J*64+65,"X(";N;")=",X
60 S$=INKEY$ : IF S$="T" THEN 30
   ELSE IF S$="U" THEN 10
   ELSE 60
```

A program lelke a 30-as sor. Itt számolja ki az (X_n) sorozat következő tagjának az értékét. A program többi része a kezdőértékek beállítását és a kiíratást vezérli.

Futtassuk ezt a programot például az $X_0 = 0.7$ kezdőértékkel. Az eredmény a *2. ábrán* látható. (Ennek vizsgálatára a cikk végén még visszatérünk.) Azt tapasztaljuk, hogy a 30-as lépéstől kezdődően már mindig ugyanazt a közelítő értéket kapjuk. Ha most kiadjuk a számítógépnek a következő parancsot:

```
PRINT COS(0.739084),
COS(0.739085),
COS(0.739086)
```

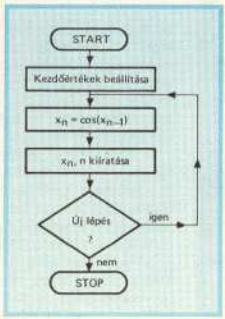
akkor a képernyőn rendre a következő számok jelennek meg:

0.739086 0.739085 0.739084

Ez azt mutatja, hogy az (1) egyenletnek e program segítségével megkapható legpontosabb megoldása a 0.739085.

Megjegyzés: Ennél nagyobb pontosságot a duplapontosságú változók használata sem ad, mivel a HT iskolaszámítógép csak a négy alapműveletet végzi el duplapontossággal.

Érdemes grafikusan is megvizsgálni az előbbi megoldást. Ábrázoljuk az $x \to \cos(x)$ és az $x \to x$ függvényeket a $\left[0, \frac{\pi}{2}\right]$ intervallumon. Az (1) egyenlet gyöke az az X_p érték, ahol a két grafikon metszi egymást. Keressük meg a koszinusz függvény grafikonján a X_0 kezdőértékhez tartozó pontot. Ezt rávetítjük az identitás függvény grafikonjára *(3. ábra)*, és az így kapott pontot levetítjük az x tengelyre. Így megkaptuk az X_1 $(= COS(X_0))$ értéket. Ezzel ugyanezt az eljárást elvégezve megkapjuk az X_3 közelítő értéket. A grafikonról jól látható, hogy ily módon valóban az (1) egyenlet X_p gyökéhez közeledünk.



1. ábra

X(1) = 0.764842	X(14) = 0.738933	X(27) = 0.739086
X(2) = 0.721492	X(15) = 0.739188	X(28) = 0.739084
X(3) = 0.750821	X(16) = 0.739016	X(29) = 0.739086
X(4) = 0.731129	X(17) = 0.739132	X(30) = 0.739085
X(5) = 0.744421	X(18) = 0.739054	
X(6) = 0.73548	X(19) = 0.739106	
X(7) = 0.741509	X(20) = 0.739071	
X(8) = 0.73745	X(21) = 0.739095	
X(9) = 0.740185	X(22) = 0.739079	
X(10) = 0.738344	X(23) = 0.73909	
X(11) = 0.739584	X(24) = 0.739082	
X(12) = 0.738749	X(25) = 0.739087	
X(13) = 0.739312	X(26) = 0.739084	

2. ábra

Megjegyzés: Ha az utolsó állítást elfogadjuk, akkor az is nyilvánvaló lesz, hogy az (1) egyenlet esetében bármilyen X_0 kezdőértéket választhatunk, mivel a $-1 \le COS(X) \le 1$ egyenlőtlenségek miatt az X_2 közelítő érték már biztosan a [0, 1] intervallumba esik.

(2) f(X) = 0

Fogalmazzuk meg most általánosan is a bemutatott eljárást, amit fixpont-iterációnak is szoktak nevezni.

Keressük a (2) egyenlet [a, b] intervallumba eső gyökét. Ha mindkét oldalához hozzáadunk X-et: $f(X) + X = X$, és bevezetjük a $g(X) = f(X) + X$, $(X \in [a, b])$ jelölést, akkor a (2)-vel ekvivalens $g(X) = X$ egyenletet kapjuk.

Tegyük fel, hogy minden $X \in [a, b]$ esetén $g(X) \in [a, b]$ is teljesül. Ekkor egy X_0 kezdőértékhez rendeljük hozzá az $X_1 = g(X_0)$, és általában az X_{n-1} értékhez az $X_n = g(X_{n-1})$ (n pozitív egész szám) közelítő értéket.

Ha az (X_n) sorozat konvergens, akkor a (2) egyenlet [a, b] intervallumba eső X_p gyökéhez tart, azaz ha $(X_n) \to X_p$, akkor $X_p = g(X_p)$, vagyis $f(X_p) = 0$.

Nézzük meg még néhány egyszerű példán a fixpont-iteráció használatát. Olyan egyenleteket fogunk vizsgálni, amelyek pontos megoldását is könnyű megadni, ezért segítségükkel ez a közelítő eljárás jól megismerhető.

(3) $X^2 - 3X + 2 = 0$

(4) $X^3 - 6X^2 + 11X - 6 = 0$

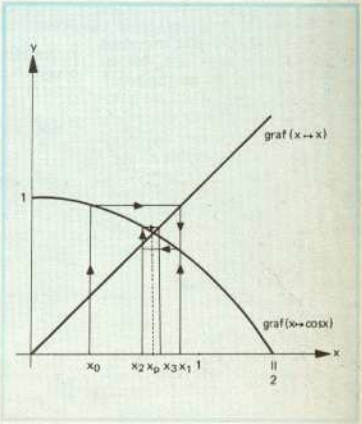
A (3) egyenlet $(X-1)(X-2) = 0$ alakba is írható, tehát két gyöke az 1 és a 2. Ha most a vele ekvivalens

$X^2 - 2X + 2 = X$

egyenletet vizsgáljuk iterációs módszerünkkel, akkor az előző programban csak a 30-as sort kell módosítani:

```
30 N=N+1 : X=X*X-2*X+2
```

A programot különböző kezdőértékekkel futtatva azt tapasztaljuk, hogy ha $X_0 \in]0,2[$, akkor néhány iterációs lépés után megkapjuk az egyik gyököt, az 1-et. Ha 0-val vagy 2-vel indítjuk az eljárást, akkor a további iterációs lépések során mindig 2-t kapunk, ami éppen a másik gyök. Ha viszont negatív, vagy 2-nél nagyobb a kezdőérték, akkor az eljárás nem lesz konvergens, egyre nagyobb és nagyobb iterációs értékeket fogunk kapni. Mindez szemléletesen is jól látható a *4. ábra* alapján.



3. ábra

A (4) egyenlet $(X-1)(X-2)(X-3) = 0$ alakba is írható. Vizsgáljuk a vele ekvivalens

$X^3 - 6X^2 + 12X - 6 = X$

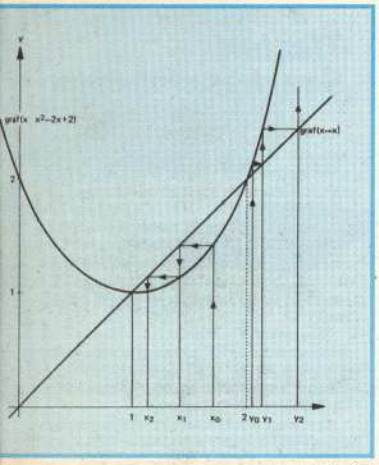
egyenletet. Ha $X_0 \in]1,3[$ kezdőértékkel indítjuk az eljárást, akkor az (X_n) sorozat 1-hez tart, ha $X_0 = 1$, illetve $X_0 = 3$, akkor minden további tag is 1, illetve 3. Az ezektől különböző kezdőértékekre pedig az (X_n) sorozat divergens lesz *(5. ábra)*.

(5) $X^2 - 2 = 0$

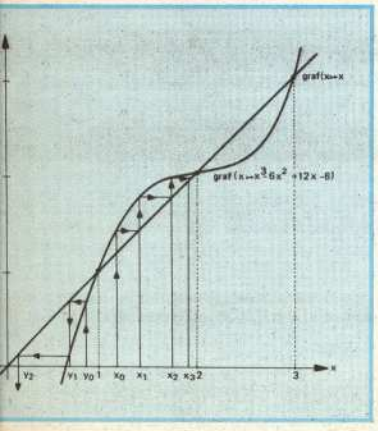
Az (5) egyenlet két gyöke a $\sqrt{2}$ és a $-\sqrt{2}$. Mindkettő irracionális szám, tehát az egyenlet pontos megoldását nem is várhatjuk a számítógéptől. Azonban ha az (5)-tel ekvivalens

$X^2 - 2 + X = X$

egyenletet vizsgáljuk az iterációs módszerünk-



4. ábra



5. ábra

kel, akkor az eddigiektől eltérő érdekes jelenséget figyelhetünk meg.

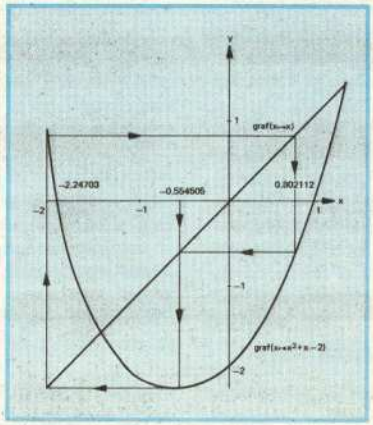
Megjegyzés: Érdemes itt a programunkat úgy módosítani, hogy a 6Ø-as sor helyett a következőt írjuk:
6Ø GOTO 3Ø
Ennek hatására a program futása felgyorsul. Ha azt rövid időre szeretnénk megszakítani, elérhetjük a SHIFT és a @ billentyűk egyidejű lenyomásával. Ezután bármelyik billentyű lenyomására a program futása folytatódik.

Ha a kezdőérték az 1, −1, Ø, ..., számok valamelyike, akkor a program által szolgáltatott értékek felváltva a Ø és a −2 lesznek. Ha azonban például Ø.5-től indítjuk az eljárásunkat, néhány iterációs lépés után a −2.24703, −Ø.554505, Ø.8Ø2112 értékek fognak ciklikusan ismétlődni. Más kezdőértékek esetében is tapasztalhatjuk, hogy az eljárás nem lesz konvergens! *(6. ábra.)* Felvetődik tehát a kérdés: megkaphatjuk-e fixpont-iterációval az (5) egyenlet valamelyik gyökét? Igen! Ehhez tekintsük például az (5)-tel ekvivalens

$$X-\frac{1}{2}(X^2-2)=X$$

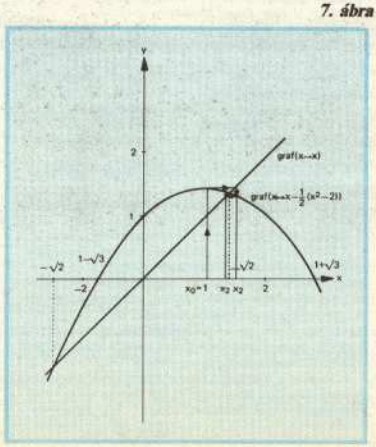
egyenletet. Ezzel végezve a fixpont-iterációt, az $X_0 \in [-\sqrt{2},2+\sqrt{2}]$ kezdőértékek esetén az (X_n) sorozat $\sqrt{2}$-höz tart, és néhány iterációs lépés után megkapjuk az 1.41421 közelítő értéket *(7. ábra)*.

Tehát ha valamely $g(X)=X$ egyenlet esetén az eljárás nem konvergens, akkor érdemes próbálkozni a (2)-vel szintén ekvivalens
$X+c\cdot f(X)=X$
egyenlettel, ahol a c alkalmasan megválasztott konstans (az előbbi példában $c=-\frac{1}{2}$ volt).



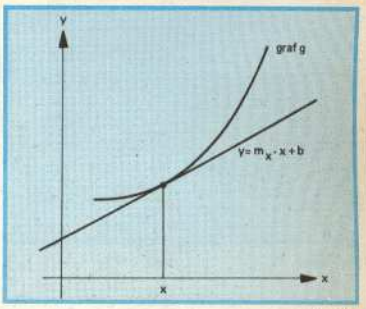
6. ábra

7. ábra

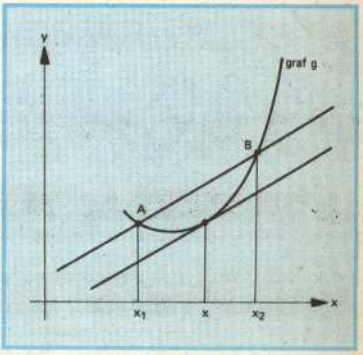


A konvergencia feltétele

A bemutatott példák és grafikonok alapos tanulmányozása során sejteni lehet, hogy az (X_n) sorozat viselkedése attól függ, hogy a g függvény „meredeksége" milyen az X_p gyök környezetében. Tegyük fel, hogy a g függvény folytonos az [a, b] intervallumon, és minden pontjában húzható itt érintő, valamint hogy



8. ábra



9. ábra

értékkészlete ezen az intervallumon része az [a, b] intervallumnak. Az $X \in [a, b]$ helyen a g függvény érintőjének a meredekségét jelöljük m_X-szel, tehát az érintő egyenlete $Y=m_X\cdot X+b$ alakú *(8. ábra)*. Ekkor igaz a következő tétel.

Legyen X_p a (2) egyenlet [a, b] intervallumba eső egyetlen gyöke. Ekkor ha létezik olyan $m<1$ valós szám, amelyre $|m_X|\le m$ teljesül minden $X \in [a, b]$ esetén, akkor az (X_n) sorozat konvergens és az X_p gyökhöz tart.

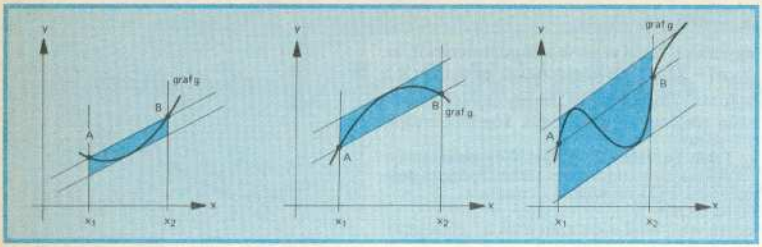
Ez szemléletesen azt jelenti, hogy ha az X_p gyök környezetében a g függvény nem „túlságosan" meredek, akkor a fixpont-iteráció konvergens lesz. Ezt a következőképpen láthatjuk be.

A bizonyításhoz felhasználjuk a Lagrange-féle középértéktételt. Legyen $X_1, X_2 \in [a, b]$, akkor létezik az $[X_1, X_2]$ intervallumban olyan X elem, amelyre

$$\frac{g(X_2)-g(X_1)}{X_2-X_1}=m_X,$$

azaz az X helyhez tartozó érintő meredeksége megegyezik az $A=(X_1, g(X_1))$ és a $B=(X_2, g(X_2))$ pontokon át húzott szelő meredekségével *(9. ábra)*.

Megjegyzés: Ezt a tételt most pontosan nem bizonyítjuk be, de a *10. ábra* segítségével szemléletesen könnyen belátható a helyessége. Tekintsük ugyanis azt a legszűkebb, az AB egyenessel párhuzamos egyenesek által határolt sávot, amely az $[X_1, X_2]$ intervallumon tartalmazza a g függvény grafikonját. Mivel a g függvény folytonos ezen az intervallumon és itt minden pontjában húzható érintő, ezért az említett



10. ábra

sáv létezik, és legalább az egyik határegyenese a g függvény kívánt típusú érintője.

A Lagrange-féle középértéktételt felhasználva minden X_{i-1} közelítő értékre teljesül a következő egyenlőtlenség:

$|g(X_{i-1}) - g(X_p)| \leq m_X |X_{i-1} - X_p| \leq$
$\leq m |X_{i-1} - X_p|,$

és a $g(X_{i-1}) = X_i$, $g(X_p) = X_p$ egyenlőségek miatt az $|X_i - X_p| \leq m |X_{i-1} - X_p|$ egyenlőtlenség. Ezt felhasználva azt kapjuk, hogy:

(6)
$|X_n - X_p| \leq m |X_{n-1} - X_p| \leq m^2 |X_{n-2} - X_p| \leq \ldots$
$\ldots \leq m^n |X_0 - X_p| \leq m^n |a - b|.$

Az $|a - b|$ egy adott érték és $m < 1$, ezért $m^n |a - b|$ az n növelésével tetszőlegesen közel kerül a nullához, azaz az X_n érték tetszőlegesen megközelíti az X_p számot. Ez pedig éppen azt jelenti, hogy az (X_n) sorozat határértéke az X_p szám.

A (6) egyenlőtlenséglláncból az is látható, hogy

(7) $|X_n - X_p| \leq m |X_{n-1} - X_p| \leq |X_{n-1} - X_p|,$

hiszen $m < 1$. Ez pedig azt jelenti, hogy ebben az esetben minden egyes iterációs lépésnél az előzőnél pontosabb közelítő értéket kapunk.

Vizsgáljuk meg, mit jelent ez például az (1) egyenlet esetében. (Az (1) egyenlethez tartozó $g = \cos$ függvény például a $[Ø.5,1]$ intervallumon teljesíti a fenti tétel feltételeit.)

Tehát a 2. ábra és a (7) egyenlőtlenség alapján látható, hogy az

X_3 közelítő érték már egy tizedesjegyre biztosan pontos, az

X_{12} közelítő érték már két tizedesjegyre biztosan pontos, az

X_{17} közelítő érték már három tizedesjegyre biztosan pontos, az

X_{22} közelítő érték már négy tizedesjegyre biztosan pontos

… stb.

Az elmondottak jól gyakorolhatók az alábbi egyenleteken:

1. $2^{-x} = x$
2. $\sin(x) = x^2$
3. $\left(\frac{10}{11}\right)^x = x$
4. $\left(\frac{1}{2}\right)^x = \lg(x)$
5. $2^x = x + 10$
6. $5^x = \sin x$
7. $2^x = 2^{-x} + \frac{10}{x}$

MAJOR ZOLTÁN

Számítástechnika a kémiában és a vegyiparban

A Tudományszervezési és Informatikai Intézet gondozásában és kiadásában a fenti címmel sorozat indult. A cím utal a sorozat eszmei tartalmára és a szerzők szándékára, hogy a számítástechnikát a vegyészmérnöki tudomány integráns részének tekintik, olyan nagy hatású módszernek, amelyet a szakember a logarléchez hasonló természetességgel kell, hogy használjon.

A kötetek egységes szerkezetűek: első fejezetük az adott terület elméletét tárgyalja a szükséges részletességgel, a második az elmélet tárgyalásán alapuló és a problémák megoldására alkalmas számítógépi programokat közöl. A harmadik fejezet az első által felvetett problémákból készített mintafeladatokat old meg a második fejezetben ismertetett programok segítségével.

Az egyes témákhoz készülő programok azonos elvek alapján íródnak, magas szintű programnyelven, olyan előírásrendszerben, amely alkalmassá teszi őket arra, hogy egy közös szervezőprogram kezelje valamennyit, és a felhasználó által kért sorrendben aktivizálja azokat. Ez az ún. CHEMISYS rendszer egyrészt maga is számítógépi program, másrészt olyan előírásrendszer, amely a szubrutinok készítői számára kötelező szabályokat tartalmaz.

A CHEMISYS rendszer programjai FORTRAN IV nyelven készültek és ESZR 22, 30, 32, 40 és 55-ös számítógépeken futnak. A programokat megkapta minden olyan egyetemi és főiskolai számítóközpont, ahol a felsorolt gépkonfigurációk működnek. A programok IBM 360 vagy nagyobb sorozatú gépeken is futtathatók.

A 7 kötetesre tervezett sorozatból Pallai Iván szerkesztésében egyelőre az alábbi öt kötet jelent meg:

Balázs–Molnár–Parti: Fluidumok szállítása és transzportjelenségek. Ára: 133,– Ft.

Vajda: Vegyipari folyamatok dinamikája és irányítása. Ára: 148,– Ft.

Fejes–Kutsán–Varga: Operációkutatási módszerek. Ára: 152,– Ft.

Kemény–Deák–Fonyó–Földes–Hunek–Láng–Rév: Elválasztási műveletek. Ára: 104,– Ft.

Tátrai–Soltész–Németh: Kémiai technológiai hálózatok. Ára: 133,– Ft.

A sorozat további két kötetének megjelenése 1985 őszére várható:

Bende–Olti: Folyadékok és gázok kémiai termodinamikája

Veress: Analitikai kémiai számítástechnika

A könyvek egyaránt alkalmasak nappali tagozatos hallgatók oktatására, posztgraduális képzésre és a gyakorló mérnök kézikönyveként. Minden kötet tartalmazza azt a felhasználó-orientált leírást, amelynek segítségével viszonylag kevés számítástechnikai ismeret birtokában is végezhetők egy adott tárgyon belül, akár részterületek és hierarchiaszintek összekapcsolásával, magas színvonalú vegyészmérnöki számítások.

A sorozat a Széchenyi István Könyv-, Katalógus és Árjegyzékboltban (Budapest V., Szt. István tér 4. 1051) vásárolható, illetve rendelhető meg.

Tisztelt Szerkesztőség!

A MTV egyik adásában olyan ismertetés hangzott el a számítástechnikai versennyel kapcsolatban, amely bennem erős aggodalmat keltett. Életemben a tanári tevékenység több, mint fél évszázadot töltött ki, s nyugalmazott egyetemi tanárként még bizonyos tanári tevékenységet ma is kifejtek. Ennélfogva sok tanítási, nevelési tapasztalat halmozódott fel tudatomban. Ezek a tapasztalatok alakították ki a nevelő és tanító munkára vonatkozó nézeteimet.

Mindenkor a jobbat akaró pedagógiai törekvések és az időálló képzési eljárások mellett foglaltam állást. Ámde ugyanakkor minden elsietett, megalapozatlan, divatmajmoló újítás ellen bírálóan nyilatkoztam.

Örömmel tapasztaltam az iskolás gyerekek számítógépek iránti érdeklődését és a számítástechnikai oktatás iskolai bevonulását –, pontosabban azt, hogy egyelőre posztgraduális képzés keretében tanárok számítástechnikai elemi képzésben részesülhettek, s tudásukat szakköri foglalkozások keretében a tanulóknak átadhatták. Örömmel hallottam, hogy iskoláink tanítási célra számítógépekhez jutottak. Valamint azt is örömmel vettem tudomásul, hogy több ezer diák vesz részt a középiskolai számítástechnikai versenyen. Mintha enyhülni látszana az ellenszenv, sőt mondhatnám közutálat, amivel a diákok túlnyomó többsége fogadja mindazt, ami matematika.

Persze, ha a matematikai középfokú műveltség szempontjából mérlegeljük a tanulók számítástechnikai ismereteit, észre kell vennünk, hogy azok nagyon kezdetleges és szűk körű matematikai elemet és gondolatot tartalmaznak. Éppen ezért támadtak aggodalmaim a MTV említett tájékoztatása nyomán, ami szerint a szóban forgó verseny első tíz helyezettjét az egyetemi felvételi vizsga matematikai részétől felmentik.

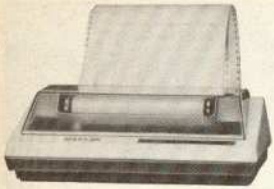
Ha ez valóban az illetékes szervek határozata, akkor azt kell mondanom, hogy nagyfokú tájékozatlanságból, felületességből fakad. Az egyetemi felvételi vizsga szerintem az a szükséges rossz, ami a mai körülményeket tekintve egyetlen szelektáló eszköz az egyetemi tanulmányokra alkalmatlan személyek idejekorán való kirostálására.

Ebből a szempontból a számítástechnikai verseny követelményei nem mérhetők össze a matematika felvételi követelményeivel.

Ne hasonlítsuk össze a számítástechnikai verseny követelményeit az ismert matematikai tanulmányi versenyek követelményeivel! Ez utóbbi verseny matematikai követelményei felülmúlják az egyetemi felvételi vizsga követelményeit. Viszont a számítástechnikai verseny *matematikai* követelményei meg se közelítik a felvételi vizsga követelményeit. Persze ez nem azt jelenti, hogy a számítástechnikai versenynek nincsenek meg a *maga sajátos* követelményei.

Én a verseny helyezettjeinek a matematika felvételi vizsga alóli felmentését indokolhatatlan, fölösleges, sőt káros *mézesmadzagnak* tartom.

KÁRTESZI FERENC
ny. egyetemi tanár,
a mat. tudományok doktora



FOK GYEM

„MARS"

MENETTELJESÍTMÉNY ADATRÖGZÍTŐ ÉS SZÁMÍTÓ RENDSZER

A korunkban uralkodó energiaválság miatt világszerte fokozódik az igény a közlekedés, ill. célfuvarozás optimális szervezésére.
A technika jelenlegi színvonalán egyre inkább csak gépi úton szerezhetők be és értékelhetők ki azok az információk, amelyek ezt a szervezési feladatot áttekinthetővé teszik.
Ehhez nyújt Önöknek segítséget a FOK-GYEM „MARS" (*M*obile *R*egistrating *S*ystem) menetíró rendszere, amely pillanatról pillanatra „fényképezi" a jármű által megtett utat és a fuvarozás alatt történt legfontosabb eseményeket, illetve lehetőségeket nyújt a mért és regisztrált adatok gyors értékelésére.
A rendszer különlegessége az, hogy a gyűjtött és regisztrált adatokat a gépjárművekben lévő menetíró készülékek rögzítik. A rögzített adatok automatikus kiértékelését, megjelenítését és további feldolgozását központi telephelyi mikroszámítógépek végzik el.
A menetíró készülékek – korszerű, nagy megbízhatóságú mikroprocesszor vezérlésű eszközök, amelyek a mérés ideje alatt a gépjárműben – kazettaszerűen helyezhetők el. A készülékek mérési programja a fuvarfeladat igényeinek megfelelően, széles határok között változtatható.
A készülékek kis méretűek, kis fogyasztásúak, mozgó elemet nem tartalmaznak. A fuvar elvégzése után a menetíró készülék a gépjárműből kivehető. A központi berendezésbe elhelyezve a regisztrált adatok a berendezés képernyőjén megjeleníthetők.
Az adatok a további kiértékelés és feldolgozás érdekében kinyomtathatók, háttértárolón is rögzíthetők, ill. eljuttathatók más ügyviteli vagy adatfeldolgozó rendszerbe.
A telephelyi berendezésekkel végezhető el a menetíró készülékek újabb fuvarútra való felkészítése is.
A rendszer szolgáltatásai az alábbi szempontokból segítik a feladatok hatékonyabb megoldását:
- gazdaságosság
- fuvarszervezés
- forgalombiztonság
- munkafegyelem.

A menetíró készülék által szolgáltatott adatok a következők.
- a fuvarfeladat, a gépjárművezető és a gépjármű azonosító adatai
- a fuvar kezdeti, befejezési időpontja, a fuvar ideje, ezen belül az állás és menetidő összesen (az állás és menetidő megkülönböztetése automatikus)
- a megtett út összesen
- a vezető által kijelölhető 6 különböző üzemmódban megtett út, állás és menetidő üzemmódonként
- üzemmódváltozások száma
- gyorshajtások összes ideje
- tápfeszültség-kiesések száma
- sebesség/idő diagram, amely az átlagsebességet rögzíti az út folyamán, az addig megtett összes út feltüntetésével. A mintavétel ideje 30 mp és 20 perc között változtatható, 30 mp-es lépésekben
- „black box", amely az utolsó 1 km-es úthoz tartozó sebességértékeket tartalmazza 4 méterenként mérve, és jelezve, hogy azon a 4 méteren történt-e fékezés
- sebesség túllépés tábla, amely a 85 legnagyobb értékű sebességtúllépés értékét és időpontját tartalmazza (a megengedett legnagyobb sebesség az azonosító adatokkal együtt adható meg)
- üzemanyag-fogyasztás
- terhelés

A menetíró készülék működése alatt a vezető számára különböző jelzéseket ad. Fénnyel jelzi a gépjármű megállását, a fékezés és a megengedett sebesség túllépésének tényét. Ez utóbbi esetben hangjelzést is adhat. Ugyancsak jelzi, ha az azonosító adatokkal való feltöltés nélkül helyezték el a gépjárműben.
A menetíró készülék mérési tartományai és pontosságai:
- az időmérés relatív pontossága: 10^{-5}
- rögzíthető állás- és menetidők üzemmódonként: 20–20 nap
- sebesség/idő diagram időtartama:
  30 mp-es felbontással – 1 nap
  20 perces felbontással – 40 nap
- sebesség/idő diagram sebességmérési pontossága: ±1 km/ó
- rögzíthető legnagyobb sebesség: 127 km/ó
- „Black box" sebességrögzítés pontossága: ±1 km/ó
- megtett útmérés pontossága: 1 m
- rögzíthető út üzemmódonként: 8000 km.

Finommechanikai és Elektronikus
Műszergyártó Szövetkezet
BUDAPEST XI., KARINTHY FRIGYES ÚT 22.
1052 Pf. 55.

HT-1080Z

Magyar nyelvű hibaüzenetek

Kezdő és haladó programozókkal egyaránt előfordul, hogy nem ismerik fel az elég szükszavú, kétbetűs angol hibaüzeneteket – keresgélniük vagy kérdezősködniük kell. Az alábbi program mellett teljes mondatos magyar hibaüzeneteknek örülhetünk.

A gépi kódú rutint BASIC segítségével POKE-olhatjuk a memóriába. A BASIC program lefutása után kitörlődik, de a gépi kódú rutin mindaddig működik, míg a gép bekapcsolva marad.

GAUTIER PÉTER

```
10 '-------------------------------------------------
20 '---          Magyar nyelvu hosszu hibauzeneteK      --
30 '--- Keszitette: Gautier Peter            KSZK     --
40 '-------------------------------------------------
50 POKE 16414,71 : POKE 16415,49 :'-- Kisbetuk bekapcsolasa -
60 CLEAR 500 :'--- StringekneK helyfoglalas ---
70 POKE 16526,248 : POKE 16527,97 :'--- Gepi rutin cime 61F8 -
80 CLS : PRINT@8*64+20,"Dolgozom!  Ne zavarj!"¡
90 C=&H4096 :'--- A programot ide pakoljuk le ---
100 BN=31 :'--- Byte szam ---
110 GOSUB 190 :'--- Hiba kiiro rutin lerakasa ---
120 GOSUB 280 :'--- Hibauzenetek lerakasa ---
130 BN=37 :'--- Byte szam ---
140 GOSUB 190 :'--- Inicializalo resz lerakasa ---
150 CLEAR 50
160 A=USR(0) :'--- A bovites elinditasa ---
170 END
180 '--- Hexa byte-ot lerako szubrutin ---
190 FOR I=1 TO BN
200 READ B$
210 B1=ASC(LEFT$(B$,1)) : B2=ASC(RIGHT$(B$,1))
220 IF B1>ASC("9") THEN B1=B1-ASC("A")+10 ELSE B1=B1-ASC("0")
230 IF B2>ASC("9") THEN B2=B2-ASC("A")+10 ELSE B2=B2-ASC("0")
240 B=16*B1+B2 : POKE C,B : C=C+1
250 NEXT I
260 RETURN
270 '--- Szoveg lerako szubrutin ---
280 READ B$
290 IF B$="END" THEN RETURN
300 IF B$="00" THEN POKE C,0 : C=C+1 : GOTO 280
310 FOR I=1 TO LEN(B$)
320 B=ASC(MID$(B$,I,1)) : POKE C,B : C=C+1
330 NEXT I
340 GOTO 280
350 '--- A hibauzenet lekezelo gepi szubrutin utasitasai ---
360 DATA E1,21,1E,43,7B,FE,2F,30,02,3E,26,0F,3C,5F,CD,07,1F,1D
    ,23,20,F9,CD,A7,28
,2A,A2,40,C3,11,1A,00
370 '--- A hibauzenetek tablazata ---
380 DATA NEXT utasitas FOR nelkul,00
390 DATA Szintaktikus hiba,00
400 DATA GOSUB nelkuli RETURN,00
410 DATA Nincs tobb adat,00
420 DATA Nem megengedett fuggvenyutasitas,00
430 DATA Tulcsordulas,00
440 DATA Nincs tobb tarolohely,00
450 DATA Definialatlan sor,00
460 DATA Tomb tulcimzes,00
470 DATA Ujra dimenzionalt tomb,00
480 DATA Nullaval valo osztas,00
490 DATA Nem megengedett felhasznalas,00
500 DATA TipusKeveredes,00
510 DATA Nincs tobb hely stringeknek,00
520 DATA A string tul hosszu,00
530 DATA Tul osszetett stringmuvelet,00
540 DATA Nem tudom folytatni,00
550 DATA Nincsen RESUME,00
560 DATA ERROR nelkuli RESUME,00
570 DATA Nem kiirhato hiba,00
580 DATA Hianyzo operandus,00
590 DATA Hianyos file,00
600 DATA  Diszkes basic utasitas,00
610 DATA END
620 '--- Az inicializalo resz utasitasai ---
630 DATA 21,00,43,22,A7,41,3E,C3,32,A6,41,21,00,60,11,00,43,01
    ,F8,01,ED,B0,21,F8
    ,44,36,00,23,22,A4,40,CD,4D,1B,C3,06,30
```

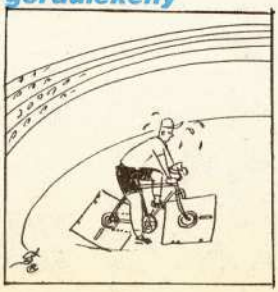
```
1 ;-------------------------------------------------------------
2 ;--------        MAGYAR NYELVU, HOSSZU HIBAUZENETEK          ---
3 ;--------          GAUTIER PETER              KSZK          ---
4 ;-------------------------------------------------------------
5 ;
6 HOSSZ   EQU     BASPR-START
7 ;
8         ORG     4300H
9 START:  POP     HL
10        LD      HL,SZOVEG
11        LD      A,E                     ;E-BEN VOLT A HIBAKOD
12        CP      2FH                     ;NEM TUL NAGY?
13        JR      C,LE
14        LD      A,26H                   ;UNPRINTABLE ERROR
15 LE:    RRCA                            ;A=A/2
16        INC     A                       ;A=A+1
17        LD      E,A                     ;E LESZ A CIKLUSSZAMLALO
18 FEL:   CALL    1F07H                   ;DATA INTERPRETER
19        DEC     E
20        INC     HL                      ;A KOVETKEZO HIBAUZENET ELEJE
21        JR      NZ,FEL
22        CALL    29A7H                   ;SZOVEG KIIRATAS
23        LD      HL,(40A2H)
24        JP      1A11H                   ;HIBAKIIRO RUTIN BELSEJE
25 ;
26 SZOVEG: DEFB   0
27        DEFM    'NEXT UTASITAS FOR NELKUL'
28        DEFB    0
29        DEFM    'SZINTAKTIKUS HIBA'
30        DEFB    0
31        DEFM    'GOSUB NELKULI RETURN'
32        DEFB    0
33        DEFM    'NINCS TOBB ADAT'
34        DEFB    0
35        DEFM    'NEM MEGENGEDETT FUGGVENYUTASITAS'
36        DEFB    0
37        DEFM    'TULCSORDULAS'

38        DEFB    0
39        DEFM    'NINCS TOBB TAROLOHELY'
40        DEFB    0
41        DEFM    'DEFINIALATLAN SOR'
42        DEFB    0
43        DEFM    'TOMB TULCIMZES'
44        DEFB    0
45        DEFM    'UJRA DIMENZIONALT TOMB'
46        DEFB    0
47        DEFM    'NULLAVAL VALO OSZTAS'
48        DEFB    0
49        DEFM    'NEM MEGENGEDETT FELHASZNALAS'
50        DEFB    0
51        DEFM    'TIPUSKEVEREDES'
52        DEFB    0
53        DEFM    'NINCS TOBB HELY STRINGEKNEK'
54        DEFB    0
55        DEFM    'A STRING TUL HOSSZU'
56        DEFB    0
57        DEFM    'TUL OSSZETETT STRINGMUVELET'
58        DEFB    0
59        DEFM    'NEM TUDOM FOLYTATNI'
60        DEFB    0
61        DEFM    'NINCSEN RESUME'
62        DEFB    0
63        DEFM    'ERROR NELKULI RESUME'
64        DEFB    0
65        DEFM    'NEM KIIRHATO HIBA'
66        DEFB    0
67        DEFM    'HIANYZO OPERANDUS'
68        DEFB    0
69        DEFM    'HIANYOS FILE'


70        DEFB    0
71        DEFM    'DISZKES BASIC UTASITAS'
72        DEFB    0
73 BASPR  EQU     $
74 ;
75 ;-------------------------------------------------------------
76 ;---  A HIBA KIIRO RUTIN KIHURKOL A RAMBA, EKKOR UGRATJUK BELE A  --
77 ;-----              MI SZUBRUTINUNKBA .                          --
78 ;-------------------------------------------------------------
79 ;
80 INIC:   LD      HL,START
81        LD      (41A7H),HL
82        LD      A,0C3H
83        LD      (41A6H),A               ;RET HELYETT JP
84 ;------      A PROGRAMOT LEMASOLJUK 6000-ROL 4300-RA     ------
85        LD      HL,6000H
86        LD      DE,4300H
87        LD      BC,HOSSZ
88        LDIR
89        LD      HL,BASPR
90        LD      (HL),0
91        INC     HL
92        LD      (40A4H),HL              ;BASIC PROGRAM ELEJE
93        CALL    1B4DH                   ;TOROL MINDENT
94 VEGE:  JP      12294                   ;KISBETUK BEKAPCSOLASA
```

```
0000  00 00 00 01 00 00 01 00 00 00 00 00 00 00 00 00  ................
10    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
20    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
30    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
40    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
50    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
60    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
70    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
80    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
90    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
A0    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
B0    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
C0    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
D0    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
E0    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
F0    00 00 00 00 00 00 00 00 00 00 37 C2 00 43 1D 02  ..........7B.C..

0100  E1 21 1E 43 7B FE 2F 38 02 3E 26 0F 3C 5F CD 07  á!.C{./8.>&.<_Í.
10    1F 1D 23 20 F9 CD A7 2B 2A A2 40 C3 11 1A 00 4E  ..# ùÍ§+*¢@Ã...N
20    45 58 54 20 75 74 61 73 69 74 61 73 20 46 4F 52  EXT UTASITAS FOR
30    20 6E 65 6C 6B 75 6C 00 53 7A 69 6E 74 61 6B 74   NELKUL.SZINTAKT
40    69 6B 75 73 20 68 69 62 61 00 47 4F 53 55 42 20  IKUS HIBA.GOSUB
50    6E 65 6C 6B 75 6C 6A 20 52 45 54 55 52 4E 00 4E  NELKULI RETURN.N
60    69 6E 63 73 20 74 6F 62 62 20 61 64 61 74 00 4E  INCS TOBB ADAT.N
70    65 6D 20 6D 65 67 65 6E 67 65 64 65 74 74 20 66  EM MEGENGEDETT F
80    75 67 67 76 65 6E 79 75 74 61 73 69 74 61 73 00  UGGVENYUTASITAS.
90    54 75 6C 63 73 6F 72 64 75 6C 61 73 00 4E 69 6E  TULCSORDULAS.NIN
A0    63 73 20 74 6F 62 62 20 74 61 72 6F 6C 6F 6B 65  CS TOBB TAROLOKE
B0    6C 79 00 44 65 66 69 6E 69 61 6C 61 74 6C 61 6E  LY.DEFINIALATLAN
C0    20 73 6F 72 00 54 6F 6D 62 20 74 75 6C 63 69 6D   SOR.TOMB TULCIM
D0    7A 65 73 00 55 6A 72 61 20 64 69 6D 65 6E 7A 69  ZES.UJRA DIMENZI
E0    6F 6E 61 6C 74 20 74 6F 6D 62 00 4E 75 6C 6C 61  ONALT TOMB.NULLA
F0    76 61 6C 20 76 61 6C 6F 20 6F 73 7A 74 61 73 00  VAL VALO OSZTAS.

0200  4E 65 6D 20 6D 65 67 65 6E 67 65 64 65 74 74 20  NEM MEGENGEDETT
10    66 65 6C 68 61 73 7A 6E 61 6C 61 73 00 54 69 70  FELHASZNALAS.TIP
20    75 73 6B 65 76 65 72 65 64 65 73 00 4E 69 6E 63  USKEVEREDES.NINC
30    73 20 74 6F 62 62 20 68 65 6C 79 20 73 74 72 69  S TOBB HELY STRI
40    6E 67 65 6B 6E 65 6B 00 41 20 73 74 72 69 6E 67  NGEKNEK.A STRING
50    20 74 75 6C 20 68 6F 73 73 7A 75 00 54 75 6C 20   TUL HOSSZU.TUL
60    6F 73 73 7A 65 74 65 74 74 20 73 74 72 69 6E 67  OSSZETETT STRING
70    6D 75 76 65 6C 65 74 00 4E 65 6D 20 74 75 64 6F  MUVELET.NEM TUDO
80    6D 20 66 6F 6C 79 74 61 74 6E 69 00 4E 69 6E 63  M FOLYTATNI.NINC
90    73 65 6E 20 52 45 53 55 4D 45 00 45 52 52 4F 52  SEN RESUME.ERROR
A0    20 6E 65 6C 6B 75 6C 69 20 52 45 53 55 4D 45 00   NELKULI RESUME.
B0    4E 65 6D 20 6B 69 69 72 68 61 74 6F 20 68 69 62  NEM KIIRHATO HIB
C0    61 00 48 69 61 6E 79 7A 6F 20 6F 70 65 72 61 6E  A.HIANYZO OPERAN
D0    64 75 73 00 48 69 61 6E 79 6F 73 20 66 69 6C 65  DUS.HIANYOS FILE
E0    00 44 69 73 7A 6B 65 73 20 62 61 73 69 63 20 75  .DISZKES BASIC U
F0    74 61 73 69 74 61 73 00 21 00 43 22 A7 41 3E C3  TASITAS.!.C"§A>Ã

0300  32 A6 41 21 00 60 11 00 43 01 F8 01 ED B0 21 F8  2¦A!.`..C.x.í°!ø
10    44 36 00 23 22 A4 40 CD 4D 1B C3 06 30 21 41 53  D6.#"¤@ÍM.Ã.0!AS
20    53 45 4D 42 4C 45 44 00 00 00 00 00 00 00 0C 9A  SEMBLED.........
30    04 1E 0C 19 00 1F 98 1D 41 73 73 65 6D 62 6C 65  ........Assemble
40    72 20 73 6F 75 72 63 65 20 66 69 6C 65 73 00 1F  r source files..
50    20 5A 38 30 20 41 73 73 65 6D 62 6C 65 72 20 20   Z80 Assembler
60    20 20 20 2F 20 0A 20 42 69 6E 61 72 79 2E 2E 2E     / . Binary...
70    0A 20 4C 69 73 74 2E 2E 2E 2E 2E 17 19 00 98 0A  . List..........
80    46 72 6F 6D 20 6C 69 6E 65 20 00 20 54 6F 20 6C  From line . To l
90    69 6E 65 20 00 0C 9A 02 9E 02 00 0C 98 0F 00 9B  ine ............
A0    13 00 20 20 20 17 00 0C 9A 03 98 32 45 72 72 20  ..      ......2Err
B0    6F 72 28 73 29 3A 20 00 1F 98 24 42 69 6E 61 72  or(s): ...$Binar
C0    79 20 6E 61 6D 65 00 0C 98 15 00 4C 41 44 52 4C  y name.....LADRL
D0    44 20 20 49 4E 43 20 44 45 43 20 41 44 43 20 53  D  INC DEC ADC S
E0    42 43 20 41 44 44 20 53 55 42 20 43 50 20 20 41  BC ADD SUB CP  A
F0    4E 44 20 4F 52 20 20 58 4F 52 20 50 4F 50 20 50  ND OR  XOR POP P
```

HIBAIGAZÍTÁS

A képszerkesztés három képsíkon című program (1985/1, 4–5. oldal) hiányosan jelent meg. A 175. sorszámú, S : DB "KEZDŐCÍM" programsortól kezdve a 191. sorszámú KIL szubrutin kezdetéig minden DB utáni szövegnek 26 karakter hosszúnak kell lennie. A záró idézőjelig tehát megfelelő számú SPACE karakter szükséges. (A szerk.)

Jó, de nem elég gördülékeny



Tiszta fejet a gépekbe!

Az idei tavaszi BNV-n járva többször is szívesen időztem a COMPUDRUG Műszaki Fejlesztő Kisszövetkezet vitrinje előtt. Hogy miért, az kiderül abból a kötetlen beszélgetésből, amelyet a szövetkezet termelési osztályának dolgozóival folytattam.

– A szövetkezet FLOTISZ DR nevű mágneslemez-meghajtó író-olvasó fej tisztítókészletét már több, számítástechnikai eszközt forgalmazó üzlet kirakatában láttam, de igazi jelentőségét csak itt, a BNV-n ismertem fel. Szeretném megtudni, hogy honnan származik az ötlet.

– Egy program fejlesztése során programozóink egy már tökéletesen „belőtt" programot többször sikertelenül próbáltak futtatni. Szoftverhibáról nem lehetett szó, hát következett a hardveres. Miután ő sem talált semmit, utolsó kísérletképpen a mágneslemez-meghajtó író-olvasó fejét tisztította meg. És a gép összeszerelése után a program hibátlanul lefutott. A következő hasonló esetben már a fej tisztításával kezdtük a javítást, ami ugyanúgy sikerre vezetett. Ekkor kezdtünk gondolkodni azon, hogyan lehetne elkerülni minden tisztításnál a floppy-meghajtó szétszedését. Így alakítottuk ki a floppy méretével teljesen megegyező tisztítókorongot, majd vegyészek segítségével sikerült megtalálni a legmegfelelőbb tisztítófolyadékot is. A készletet több nagy számítógépgyártó, forgalmazó és szervizelő vállalat tesztelte.

Egy készlet két tisztítófloppyt és egy spray-t tartalmaz. A spray-vel bepermetezett, speciálisan impregnált korongot a floppy-egységbe kell helyezni, és a használati utasításban megadott rövid programmal 30 másodpercig olvasásra kell késztetni a gépet. Ez alatt a rövid idő alatt a fej tökéletesen megtisztul a rárakódott szennyeződéstől. Ezt a műveletet átlagosan igénybe vett gépeknél havonta egyszer kell elvégezni. Egy FLOTISZ készlet egy meghajtóhoz kb. 2 évig használható.

– Az író-olvasó fej tisztítása hatással van a floppyra is?

– Természetesen, hiszen a bepiszkolódott fej károsíthatja a lemezt is, így a FLOTISZ használata a mágneslemezek élettartamát is növeli.

– Milyen típusú gépekhez használható a FLOTISZ?

– Minden olyan géphez, amely egy- vagy kétoldalas, 5 1/4 vagy 8"-os floppyt használ.

– Milyen a FLOTISZ eddigi piaci sikere?

– Külföldön nagy sikerrel forgalmazzuk. A tavaszi tokiói vásáron japán szerződést kötöttünk eladására, a hannoveri vásáron pedig 8 ország kért bemutató tesztelést, amelyek jelenleg is folynak, és szép eredménnyel kecsegtetnek.

– A vitrinben más FLOTISZ termékeket is láttam.

– A FLOTISZ DC tisztítókészlet mágneslemez-csomagok tisztítására használható. Három másik, szintén új fejlesztésünk a számítástechnika körén kívül is alkalmazható. A FLOTISZ HC magnetofonok és mágnesszalagos tárolók mágnesfejének tisztítására szolgál. A FLOTISZ TC karos, gömbfejes és margarétafejes írógépek és nyomtatók tisztítására alkalmas. A család eddig utolsó tagja, a FLOTISZ SC készlettel televízió, monitor és terminálképernyők tisztíthatók; nemcsak számítóközpontokban, hanem otthon is jól használható.

– Kérem beszéljenek új termékcsaládjuk első két tagjáról is; azt hiszem, a vitrin körül látható nagy érdeklődés ezeknek is szól.

– Standix LT_1 és LT_2 néven két azonos funkciójú, csak formai kialakításukban eltérő leporellótartó állványt fejlesztettünk ki. Az állvány tárolja a behúzandó printerpapírt és a nyomtatóból kijövő nyomtatott anyagot, segíti a folyamatos printerpapír-ellátást, sérülésmentesen összegyűjti a nyomtatványokat, és egyszerű szétválogatást tesz lehetővé. Emellett könnyen szállítható, szétszedhető, egyszerűen állítható újra össze. A polcok magassága a feladattól és a nyomtató elhelyezésétől függően tetszőlegesen állítható.

– A BNV-n nagy sikerrel szerepeltek termékeik, ezt bizonyítja az OKISZ díja is. Reméljük, a magyar számítógépfelhasználók körében mielőbb elterjednek ezek a termékek, amelyek a számítógéppark jobb kihasználását segíthetik elő.

(-s-h)

Alapozás XI.

Pneumatikus operátorunkkal egyszerű kapcsolások révén számos hozzárendelést, transzformációt, kapcsolatot sikerült műszaki formában hasznosíthatóvá tennünk. E hasznosítás lehetőségeivel foglalkozunk most, és módszert adunk arra, hogy hogyan lehet adott függvényekhez tartozó kapcsolásokat (operátorcsapatokat) szerkeszteni.

A függvény és technikai megvalósítása

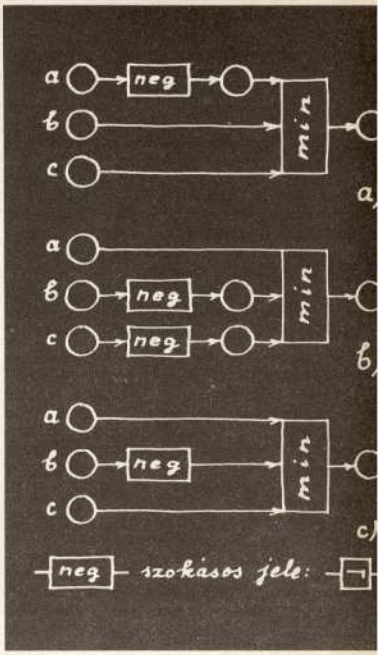
Olyan függvények modellezésére láttunk példákat, amelyeknek minden bemeneti és minden kimeneti változója két értéket vehet fel. A matematikai függvény három alkotórészes, időtől független fogalom; az értelmezési tartomány, az értékkészlet és e két halmaz elemei közötti kapcsolatok definiálják.

A matematikai függvényben mindig rendelkezésre áll az összes „függetlenváltozó-függőváltozó érték" pár, ami a függvényben egymáshoz van rendelve, amit összetartozónak tekintünk. Ez az egyidejű rendelkezésre állás azonban a gyakorlatban nagyon ritka jelenség. Leggyakrabban az a helyzet, hogy adott függetlenváltozó értékhez meg kell keresnünk, vagy ki kell számítanunk a hozzá tartozó függőváltozó értéket. Tehát egyetlenegy egymáshoz rendelt elempár (értékpár) sem áll rendelkezésünkre időben korlátozatlanul, időtől függetlenül. Sőt még az is igaz, hogy nagyon gyakran a szóban forgó matematikai függvény egyetlenegy, konkrét, egymáshoz rendelt elempárja sincs birtokunkban egyidejűleg, mert a függetlenváltozó értéket felhasználjuk (elhasználjuk) a függőváltozó érték kiszámításához, oly módon, hogy az fizikailag is eltűnik. A függvénymodellezésnél tehát nem matematikai függvényt modelleztünk, hanem függvényhelyettesítésiértékképzési műveletet, melynek során egyszer egyidőre felbukkan a függetlenváltozó érték és egyszer egyidőre a függőváltozó érték. Hogy ezek mikor és meddig elérhetők számunkra, az más kérdés.

Láttunk olyan pneumatikus rendszereket, amelyeknek bemeneti pontjain a függetlenváltozó értékeit beállítva, egy bizonyos idő múlva megjelentek a kimeneti pontokon egy függvény által a bemeneti állapotokhoz (értékekhez) rendelt állapotok (értékek). Számos ilyen függvényhelyettesítésiérték-képző operátort, operátorcsapatot szerkesztettünk. A sokféleség láttán természetesen vetődik fel a kérdés az előállíthatóság korlátaira vonatkozóan. Képesek vagyunk-e, a már megismert módon, minden olyan függvény helyettesítési értékét előállító operátorcsapatot létrehozni, amelynek minden függő és minden független változója csak két értéket vehet fel? *A válasz igenlő.* Ennek a rendkívül nagy jelentőségű igazságnak a bizonyítását elég olyan függvényekre elvégezni, amelyeknek egyetlen kimeneti változójuk van csak. Ha ugyanis minden ilyen függvény esetében igaz a szóban forgó állítás, akkor mivel minden *n* kimenetű függvény *n* darab 1 kimenetű függvényből összeállítható, kijelentésünk minden n kimenetű függvényre is igaz lesz. Feladatunkat konstrukcióval oldjuk meg. Megadunk egy eljárást, amellyel minden egykimenetű végessokváltozós bináris függvényhez tudunk annak helyettesítési értékét kiszámító operátorcsapatot (kapcsolást) szerkeszteni.

	operandus			operátum
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	1
8	1	1	1	0

1. ábra



2. ábra

Speciális operátorcsapatok

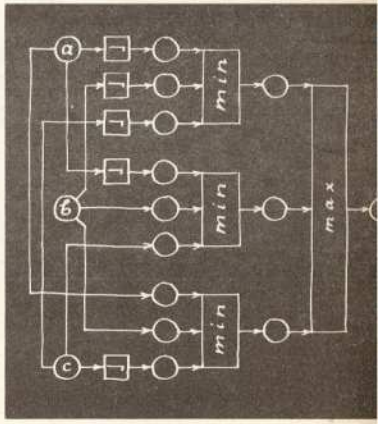
Tételünk bizonyítását könnyű volna – teljesen egzakt módon – a bemeneti változók száma szerinti indukcióval elvégezni. Ehelyett egy szemléletesebb utat választunk. Konkrét példán mutatjuk meg a bizonyítás gondolatmenetét.

Legyen példánkban a bemeneti változók száma három!

Legyenek az operandusok *a, b, c,* a kimeneti operátum pedig *k*!

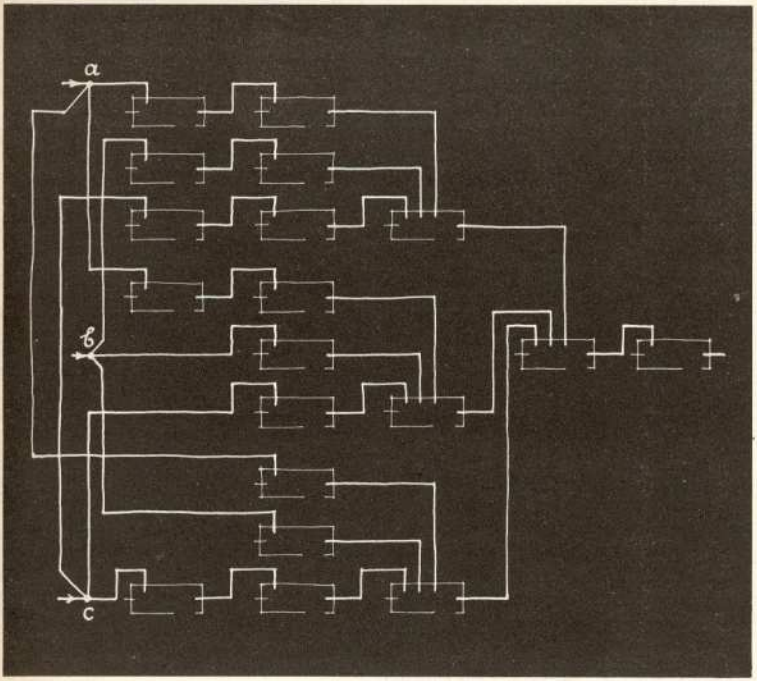
Legyen továbbá az operandus, operátum táblázat a következő! *(1. ábra)*

A táblázat minden sorához szerkesztünk egy speciális operátorcsapatot. A 4., 5. és 6. sorhoz például a *2. ábrán* látható kapcsolást konstruáltuk. Ne felejtsük, minimumképzőt, maximumképzőt és negálót pneumatikus operátorokkal már tudunk építeni. (A negáló olyan operátor, amelynek kimenete – durva modellje szerint – akkor és csak akkor „nem fúj", ha a bemenete „fúj".) E csapatocskák szerkesztési szabálya az, hogy a kimeneti változó mindig olyan hárombemenetű minimumképző kimeneti pontja, amelynek bemeneteire mindig *a, b,* és *c* csatlakozik közvetlenül vagy egy negálón keresztül, aszerint, hogy a táblázatban 1 vagy 0 van az illető változó oszlopában.



3. ábra

Nyilvánvaló, hogy egyforma sorokhoz egyforma operátorcsapatocskák tartoznak. És látni fogjuk azt is, hogy különböző sorokhoz különböző felépítésű és különböző módon működő csapatocskák tartoznak. Ez a nyolc csapatocska nagyon érdekes viselkedésű. Ha ugyanazt az állapothármast (értékhármast) adjuk bemenetükre, akkor a nyolc közül pontosan egynek a kimeneti értéke biztosan 1 lesz, a többié



4. ábra

pedig 0 (bizonyos idő múlva, bizonyos ideig). (A pontosság kedvéért nélkülözhetetlen 3 „bizonyos idő" írásbeli hangsúlyozásától most – az egyszerűség kedvéért – eltekintünk, a gondolati elhanyagolás azonban súlyos hibákra vezet, ezért az olvasó sose feledkezzen meg róla.) Mindig lesz tehát egy operátorcsapatocska, amelynek kimenetén az 1 érték lesz, több ilyen pedig sohasem fordulhat elő.

Lássuk ennek bizonyítását!

Mivel mindegyik bemenet 2 állapotban lehet, és három bemenet van, az összes lehetséges egymástól különböző bemeneti állapot darabszáma legfeljebb kétszer kétszer kettő, azaz 8 lehet. De mivel 8 különböző állapot biztosan van (lásd táblázatunk sorait), ezért az összes lehetséges egymástól különböző bemeneti állapot száma legalább 8. Ebből következik, hogy az egymástól különböző bemeneti állapotok (az állapotváltási állapotokat természetesen nem számítva) pontos száma nyolc.

Felhívjuk a figyelmet az előbbi bizonyítási fogás fontosságára. Az egyenlőség bizonyításának tulajdonképp ez az általános módszere. Így például 2 halmaz akkor egyenlő, ha egyik nem bővebb és nem is szűkebb mint a másik. Két szám akkor egyenlő, ha az egyik legfeljebb akkora, de legalábbis akkora, mint a másik. Különösen fontos ez a módszer a kombinatorikus feladatok esetében, e téren ugyanis elterjedt divat hiányos bizonyítások közlése.

Azt, hogy táblázatunk sorai mind különböznek egymástól, azaz nincs közöttük két egyforma, a következőképp láthatjuk be.

A táblázat sorai rendezve vannak. Először a 0 kezdőeleműek, majd az 1 kezdőeleműek vannak felsorolva. Az azonos kezdőeleműeken belül, először a 0 középső eleműek, majd pedig az 1 középső eleműek következnek. Végül pedig az azonos kezdő- és középső eleműek közül a 0 utolsóelemű megelőzi az 1 utolsóeleműt. Ha a táblázatban volna két különböző sor, amelyben ugyanaz a számsorozat van, akkor e két különböző sornak vagy a táblázat felső vagy alsó felében kell lennie, hiszen első jegyük is megegyezik. Tegyük fel, hogy a felső felében van a két sor. A második jegy azonossága miatt a felső félnek vagy a felső vagy az alsó felében kell lennie a két különböző helyre írt, de azonos tartalmú sornak. Tegyük fel, hogy a felső fél felső felében vannak e sorok. Itt viszont csak 2 sor van, amelyek viszont biztosan különbözők, hiszen különböző az utolsó elemük.

Táblázatunkban tehát az összes (különböző) bemeneti állapot-lehetőség szerepel egyszer és csakis egyszer (más szóval pontosan egyszer). Nyilvánvaló, hogy egy adott sorhoz tartozó operátorcsapatocskák kimenetén, ha a bemenetén a táblázatbeli sorában szereplő állapot van, biztosan 1 érték lesz. Egy minimumképző operátor kimenetén ugyanis akkor és csakis akkor (más szóval pontosan akkor) van 1 érték, ha minden bemeneti változójának értéke 1. Operátorcsapatocskánkat viszont pont úgy szerkesztettük, hogy ez így legyen. Ha ugyanis valamelyik bemeneti változó értéke a táblázat szerint 0, azt negáltuk és ezáltal biztosítottuk, hogy minden bemeneti érték 1 legyen. Bármilyen bemeneti állapot esetében mindig van tehát legalább egy operátorcsapatocska, amelynek kimenetén az 1 érték van. Több azonban nincs. Tegyük fel ugyanis, hogy van olyan bemeneti állapot, melynél van 2 darab egymástól különböző, 1 kimeneti értéket adó operátorcsapatocskánk. Ezek mindegyikének kimenete egy minimumképző kimenete. E minimumképzőnek kimenete – feltevésünk értelmében – 1 állapotban van. Tehát mindegyik bemeneti változójának is 1 állapotban kell lennie (helyesen: kellett lennie bizonyos idővel előbb...). Ha *a* értéke 1, akkor ez a pont mindkét csapatocskában közvetlenül csatlakozik a minimumképző első bemenetére. Ha *a* értéke 0, akkor mindkét csapatocskában ez a pont, (ill. az az operandus, melynek ez a kimenete) negáltan csatlakozik a minimumképző első bemenetére, különben a minimumképző kimenete 0 állapotban lenne. Hasonlóan okoskodhatunk a *b* és a *c* változók esetében is. Tehát a két operátorcsapatocskának teljesen egyformának kellene lennie, és így volna 2 egyforma operandus-hármas is a táblázatban, ez azonban lehetetlen. Így tehát igaz, hogy különböző sorokhoz különböző operátorcsatapocskák tartoznak. Megállapíthatjuk ezek után, hogy minden operátorcsapatocskának különböző, és egy konkrét bemeneti állapotnál egyik közülük biztosan 1, a többi pedig biztosan 0 kimeneti értékű. (Az lesz közülük 1 kimeneti értékű, amelyet a szóban forgó bemeneti állapot sorából szerkesztettünk.)

Csatoljuk azoknak az operátorcsapatocskáknak a kimeneti pontjait egy maximumképzőre, amelyek sorában az utolsó oszlopban a táblázatban 1 van! Ezáltal kialakul egy operátorcsapat, amely meg fog felelni követelményeinknek, azaz az operandus-operátum táblázatnak megfelelően fog működni. Kimeneti változója ugyanis 1 értékű lesz, minden olyan esetben, amikor az operandus-operátum táblázat *k* oszlopában 1 áll, hiszen az ilyen sorokhoz tartozó operátorcsapatocskák kimenetén, és ezáltal a maximumképző bemenetén is, ilyenkor és csak ilyenkor 1 áll. Ha viszont olyan bemeneti állapotról van szó, amely mellett *k* értéke 0, akkor a maximumképző összes bemenetére az őhozzá kapcsolt operátorcsapatocska zérust juttat, hiszen ilyen esetben egy olyan operátorcsapatocska kimenetén lesz 1, amely nem befolyásolja a maximumképzőt, mivel nincs vele kapcsolatban, hiszen ezeket az operátorcsapatocskákat nem is juttattuk szerephez, nem vettük be a csapatba.

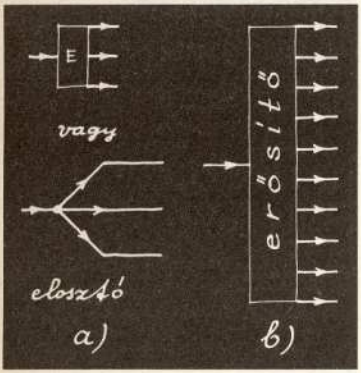
Egzisztencia és unicitás

Az imént bizonyított (helyesebben szemléltetett) felbecsülhetetlen jelentőségű alaptételünk ún. *egzisztencia tétel*, magyarul: *létezést kimondó tétel*. Az egzisztencia tételekkel kapcsolatban mindig meg szoktuk vizsgálni az *unicitást* is, azaz azt, hogy ha van a szóban forgó valamiből, akkor hány van. Ha *pontosan egy*, akkor beszélünk *unicitásról*. Ha *egzisztencia is és unicitás is van*, akkor azt úgy fejezik ki, hogy *„létezik egy és csakis egy..."* Az ilyen tételeknek feladatmegoldásokban az a haszna, hogy, ha valahogyan találunk *egy* megfelelő megoldást, akkor arról biztosak lehetünk, hogy az *a* megoldás, nem kell többértelműségből eredő problémákkal foglalkoznunk.

Egzisztenciánk tehát van. Van-e unicitás? Azaz igaz-e, hogy minden függvényhez egy és csakis egy operátorcsapat szerkeszthető? Ez nem igaz. (Jóból is megártana a sok.) *Minden függvényhez több, sőt minden függvényhez végtelen sok helyettesítésiérték-kiszámító operátorcsapat létezik.* Ezek feltérképezésével nem foglalkozunk, mert nekünk mindig csak egy megoldásra lesz szükségünk a következőkben, és hogy az milyen, az most nem lesz fontos számunkra.

Elosztás, erősítés és terhelhetőség

Megrajzolva operátorcsapatunkat, feltűnik, hogy *a*, *b* és *c* mindegyikére három vezeték, három operátor csatlakozik *(3. ábra)*. Az *a*, *b* és *c* operanduspontokra csatlakozó operátorok száma ennél is több lenne, ha feladatunk nem 1 kimenetű operátor szerkesztése lenne, hanem

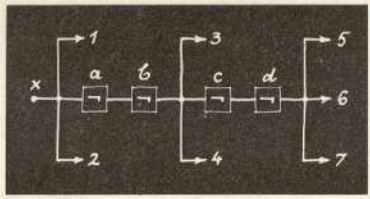


5. ábra

több kimenetűé. Ne felejtsük el, hogy több kimenetű operátorunkat kimenetként, külön-külön valósítjuk meg, és e megvalósításokat mind ugyanazokra az operanduspontokra kell csatlakoztatnunk. (Ezt lehet más módokon is csinálni, ezekkel azonban most nem foglalkozunk.)

Megrajzolva a pneumatikus kapcsolást *(4. ábra)* – ez egyszerűsíthető, az egyszerűsítést végezze el az olvasó – az operanduspontokról több operátort kell „szusszal" ellátni. Természetes kérdés, hogy a légáram felezése, harmadolása stb. meddig mehet. Meddig „bírja szusszal" egy operátor kimeneti pontja? *Mi a maximális számú operátor, amely egy pontra csatlakoztatható?* Mi a teljesítőképessége, mi a *terhelhetősége* egy pontnak? Világos, hogy *nem végtelen*. A légáram osztásánál, amit elosztó operátorokra bízhatunk *(5. ábra)*, ha ezek nagyon sokfelé végeznek elosztást, egy-egy „adag" oly jelentéktelenné válhat, hogy nem képes a pneumatikus operátor tápsugarát eléggé megzavarni, és így hatása bizonytalanná vagy elégtelenné válna.

Egy-egy operátor terhelhetősége 3, 4 vagy 5 szokott lenni. Azaz ennyi másik operátor bemenete csatlakoztatható kimenetére. A terhelhetőségi szám azonban sohasem haladhatja meg a 10-et. Mi a teendő az olyan esetekben, amelyekben például 20 operátort kell „meghajtanunk" egyetlen kimeneti pontról. Két megoldási lehetőséget említünk. Az egyik a gyakorlati út, az ún. erősítő operátorok használata. Az erősítő olyan berendezés (operátor), amelynek sok kimenete van (például 10), és a bemeneti állapottal egyenlő állapotot mind a 10 kimenetére változatlan módon szolgáltatja. Természetesen energiát használ fel, és a kimenetei táplálásához szükséges levegőt nem az egyetlen bemenetéről veszi, mint a közönséges elosztó (5.b ábra). Vegyük észre, hogy az elosztó és az erősítő nem különbözik egymástól lényegesen. Szerepük a *gyakorlati megvalósítás szempontjából nélkülözhetetlen*, elméletileg azonban elhanyagolható. Ez az oka annak, hogy elvi kapcsolási rajzokon ritkán szoktak szerepelni.



6. ábra

Az erősítés más módja a következő, amely egyúttal módszer is arra vonatkozóan, hogy hogyan készíthetünk erősítőt. Tegyük fel, hogy operátoraink kimenetének terhelhetősége 3, és egy operátorról 6 másikat kellene vezérelnünk, és nincs erősítőnk. A szóban forgó operátor kimenetéről *(6. ábra)* 3 részre osztva a légsugarat, 2 sugárral 2 vezérelendő operátort vezérlünk. A maradék 1/3 résszel két egymás után kapcsolt negálót (a, b) vezérlünk, melynek kimeneti légsugarát szintén három részre osztva, 2 újabb vezérelendő operátor vezérlését oldhatjuk meg (kicsit késve!). A maradék 1/3 résszel ugyanúgy járunk el, mint előbb. A sorba kapcsolt negálók (*c* és *d*) 3 darab kimenetén ugyanaz az állapot lesz, mint az x ponton volt bizonyos időtartammal előbb. Így az *x*-beli állapotot kis késéssel 2, 2, majd 3 helyre tudtuk eljuttatni.

Ne felejtsük el, hogy az erősítő és elosztó működéstörvényében ugyanúgy szerepe van a „bizonyos időnek", mint más operátorok működésében. E szerepek pontos figyelembevételét majd megfelelő pontosságú matematikai modellek birtokában fogjuk elvégezni.

Befejezésül felhívjuk a figyelmet egy gyakorlati feladattípusra. Készítsünk kapcsolásokat minimumképző, maximumképző és negáló (azaz 0-ból 1-et, 1-ből 0-t csináló vagy más szóval 1/2-re tükröző) operátorokkal adott operandus-operátum táblázatokhoz, és építsük fel e kapcsolásokat részletesen, pneumatikus operátorokat használva.

Összefoglalva az eddigieket, most már abban a helyzetben vagyunk, hogy a legfontosabb számológép-építő elemek közül a kódoló, dekódoló, multiplexer, demultiplexer és az aritmetikai műveleteket elvégző elemeket néhány szóval el tudjuk majd intézni. Ezek után már csak a tárelemek, a kezelőszervek (kapcsolók, nyomógombok) és a kijelzők tárgyalására van szükség, és ezzel minden digitális elven működő berendezés belső „hardver" szempontú feltérképezését elvégeztük.

Továbbhaladásunk következő lépése operátorainknak az eddig mellőzött „bizonyos idők" alatti viselkedésének pontos feltérképezése lesz. Különös és érdekes dolgok történnek a „bizonyos idők" alatt az operátorcsapatokban, amelyeknek pontos ismerete adja majd kezünkbe a lehetőséget nemcsak a megbízható rendszerek építéséhez, hanem tárelemek konstruálásához is.

POGÁNY CSABA

Havass Miklós a Számítástechnika-alkalmazási Vállalat vezérigazgató-helyettese. A 45 éves vezetőszakember a szegedi egyetem matematika szakán szerzett tanári és alkalmazott matematikusi diplomát. Diplomamunkáját a számítógépes zenéről írta. Részt vett az első nagy hazai számítógép felállításában, a NIM-ben. Tevékenysége fokozatosan a számítástechnika szoftverterületén bontakozott ki, ebben a témában több publikációja is megjelent. Munkája mellett a Neumann János Számítógéptudományi Társaságban a szoftver szakosztály elnökeként vállal aktív szerepet. A SZÁMALK-nál jelenlegi beosztásában az oktatás, illetve minden szoftverrel kapcsolatos vállalati tevékenység felügyelete és a szoftver kutatás-fejlesztés koordinálása a fő feladata.

– Ön régebben a számítástechnikai szakmában szoftver szakemberként volt ismert és elismert, jelenleg viszont egy 1300 fős nagyvállalat vezérigazgató-helyettese. Mint vezető, nem került-e szembe korábbi önmagával, szemléletével, vagyis azzal, hogy azelőtt a szakma gondjait saját bőrén érezve, „csupán" szakemberként látta a dolgokat?

– Annak idején kutatóként teljesen azonosultam az elérendő célokkal, csak az adott feladatra koncentráltam, és így rendkívül ingerelt, ha hiányoztak a feladatokhoz szükséges eszközök. Ezekből pedig a vezetés különböző szintjein egyre többel gazdálkodhattam. A megnövekedett lehetőségek reményében fogadtam el a megbízásokat. Ami a kérdés szűkebb értelmezését illeti, úgy érzem, nem változtam meg attól, hogy vezető lettem. Amit két évtizede vérlázító igazságtalanságnak tartottam, azzal ma sem vagyok kibékülve, mindössze arról van szó, hogy egy-egy dolog megítéléséhez az adott tényen túlmutató összefüggéseket is megpróbálom figyelembe venni. Példával élve: ha a vonalemberekének, vagyis Menő Manónak elfogy a vonala, és gödörbe esne, felháborodásában izgatottan pörli az egész világot. Jelenlegi beosztásomban sokféle vonalra és gödörre van rálátásom.

– Mondana egy ilyen nagyobb „gödröt"?

– Szempontunkból talán a legnagyobb,

A számítástechnika nem játék!

BESZÉLGETÉS HAVASS MIKLÓSSAL

hogy a különféle fejlesztési programok elsősorban az eszközök gyártását „nyomják", mindent alárendelve ennek, sajnos a felhasználók érdekeit is. Az alkalmazóknak az lenne a jó, ha a gyártás inkább az ő igényeikhez alkalmazkodna. A mostani gyakorlatnak köszönhető, hogy az alkalmazási kultúrák még nem épültek be kellő mértékben.

– Eszerint a gyártók és a felhasználók szemben állnának egymással?

– Ha a megfogalmazás kissé sarkított is, nincs messze az igazságtól.

– Hogyan jelentkezik ez a személyi számítógépek esetében?

– A gyártók a hozzáférhető olcsó mikroelektronikai eszközökből kisipari módszerekkel 100–150-féle gépet raknak össze. Csak összehasonlításul említem Csehszlovákiát, ahol mindössze 4–5-fajta gép van forgalomban. Hazai árviszonyaink nem teszik lehetővé, hogy piaci értékítélet alapján a legjobbak kiváljanak, és ezeket nagy szériában, olcsón lehessen gyártani.

– Számomra ez elég nehezen felfogható. Azt sem értem, hogy érhetjük el ilyen körülmények között azt az egyre nyomatékosabb gazdasági kényszerrel amúgy is sürgető célkitűzésünket, hogy külső piacra vitt termékeinkben minél több mikroelektronikai tartalom jelenjen meg?

– Gazdasági struktúránk korábban nem volt túlzottan piac-orientált. Hogy nyíltabban kell gazdálkodni, azt már régebben felismertük, erre mutat a gazdasági mechanizmus reformja. Tudom, sokan elégedetlenek az átállás ütemével, hurrá-optimizmusra szerintem sincs okunk. E beszélgetés keretét azonban meghaladja a gátló tényezők felsorolása.

– Mégis, az ismert külgazdasági helyzetben hogyan tartja lehetségesnek az előrelépést, illetve a gyakran hangoztatott elmaradás felszámolását a számítástechnika eredményeinek gyorsabb ipari kihasználásában?

– Az elmaradás relatív fogalom. A számítástechnika a fejlett országokban is az utóbbi évtized második felében mutatott jelentősebb fejlődést, és talán nem túlzás azt állítani, hogy a szakma viszonylag gyorsan reagált. Erre a mostani vásáron is láthattunk előremutató példákat néhány „exportnak kitett" vállalatunknál.

– Ez azonban egyelőre sajnos még nem jellemző. Véleménye szerint mire számíthatunk a jövőben?

– Népgazdaságunknak elsősorban azokon a területeken kell produkálnia, ahol nem kell sok energia és nyersanyag, de szellemi tőkénk hatékonyan felhasználható. Ilyen terület, ahol a nemzetközi piacon is a legtöbbet érhetjük el, a szoftverkészítés. Az utóbbi időben Kuvaittól a Wall Street Journalig rengeteg cikk beszél méltatóan a magyar szoftverekről, bizonyítékképpen, hogy már sikerült a széles külpiacokon is megjelennünk.

– Hogyan lehetne ezt a folyamatot a lehető legnagyobb mértékben kiszélesíteni, méghozzá minél gyorsabban?

– Nekem kezdettől fogva szilárd meggyőződésem volt, hogy a számítástechnika nem játék, így a szoftver sem az! Komoly technológiai feltételek, eljárások beépítése fegyelmezett kollektív munkát igényel. Nálunk valamilyen oknál fogva az egyes emberek és egyes cégek még nem tanultak meg igazán együttműködni. Mindenki többé-kevésbé jól csinálja a maga dolgát, nem törődve azzal, hogy a kooperáció jelentősen több sikert hoz, mint ha egyszerűen összeadjuk a különkülön elért eredményeket. Ha például én, mint SZÁMALK csinálok egy takarmányoptimalizáló programot, egy másik cég egy tehéngenetikai rendszert, egy harmadik ismét külön állattenyésztési költséganalízist, sokkal kevesebbet csináltunk, mint ha összeálltunk volna egy összefüggő, nagy rendszer létrehozásáért.

Az együttműködési szellem hiányzik, ezért nem épülhetett még be eléggé a számítástechnika a köztudatba, nem válhatott a kultúrközeg természetes elemévé. Ide tartozik és ugyancsak az együttműködési hajlam hiányával magyarázható a mikrogépek már említett burjánzó sokfélesége. A gyártók nem vették figyelembe, hogy az eszközök hirtelen jutnak el túl sok felhasználóhoz, akik ráadásul zömében laikusok. Ezekhez legfeljebb kisszámú specifikus feladatra alkalmas szoftver készíthető, hiány van írásos anyagokban is. Nincs egy fórum, ahol a személyi számítógép tulajdonosok elmondhatnák észrevételeiket, hogy megvalósíthatók legyenek a szükséges változtatások. Ez sem kedvez a számítástechnika közkinccsé válásának. Nem hangsúlyozhatom eléggé, hogy az ügy kis és nagy méretekben egyaránt csapatmunkát kíván.

– Ez a szakmának szóló üzenetként is értelmezhető?

– Pontosan. Azzal kiegészítve, hogy a hatékony csapatmunka olyan humánus értékek hangsúlyos érvényesülését feltételezi, amelyek belülről motiválóan hatnak abba az irányba, hogy az egyének természetes módon rendeljék alá tevékenységüket a közösség érdekeinek.

LACZKA MIKLÓS

A Comput–80 mikroszámítógép

A Comproject Számítástechnikai és Automatizálási Mérnöki Iroda GM által kifejlesztett gépcsalád a Z80 alapú és CP/M 2.2 rendszerrel kompatibilis mikroszámítógépek egyike a hazai piacon. A gyártási jog megvételével a Villamosberendezés és Elektronikai Vállalatnál (VBKM) is megkezdődött a gépek előállítása. A tanácsigazgatás mikroszámítógépes igényeinek kielégítésére kiírt pályázat egyik nyertese ez a géptípus, így mindenképpen várható, hogy elterjed hazánkban.

Moduláris hardverfelépítés

A Comput–80 egyik sajátossága a moduláris kialakítás. Ennek központi eleme az STDX sínrendszer, amely fizikailag az áramköri kártyarendszert magába foglaló műszerfiók nyomtatott áramköri hátlapjaként van kialakítva. A kártyák kisméretű, ún. Európa-kártyák, amelyek adott hardverfunkciókat látnak el.

Az eddig gyártott rendszerek többsége (Comput–80/20, /30 és /40) olyan egyműszerfiókos és egymunkahelyes kialakítású, hogy a nagyobb külső méretű, 8 hüvelykes hajlékony mágneslemezegységek és a színes megjelenítő alrendszer kivételével valamennyi perifériális egység egy zárt készülékházban helyezkedik el. Az ilyen rendszerek maximális hardver-konfigurálhatóságát mutatja *ábránk*.

A központi egység (CPUC) 2 vagy 4 MHz-es órafrekvenciájú Z80 processzorral, 8 kbájt EPROM-mal és 64 kbájt operatív memóriával rendelkezik. A típustól függően 10, 19, 27 vagy 40 Mbájt nem formátumozott kapacitású, 1 db beépített Winchester-mágneslemez (5 ¼ hüvelykes) vezérlését külön kártya végzi, amely egy általános perifériainterfész kártyán (SASI) keresztül csatlakozik a rendszerhez. A hajlékony mágneslemezegységek vezérlőkártyája (FPYW) maximálisan 4 db szoft-szektoros szervezésű meghajtó (5 ¼" vagy 8") kezelésére képes. Az 5 ¼ hüvelykes japán gyártmányú meghajtók félmagas kivitelűek, és kétféle változatban állnak rendelkezésre. Az egyik változatban 250 kbájt a nem formátumozott tárkapacitás, a másikban 1 Mbájt. A 8 hüvelykes meghajtók MOM gyártmányúak, nem formátumozott kapacitásuk 500 kbájt, és fizikailag különálló egységben vannak elhelyezve (Compflex).

A monitorvezérlő (CRT) alfanumerikus üzemmódban 24 vagy 26 sort jelenít meg, soronként 80 karakterrel. Kvázigrafikus üzemmódban egy karakterhelyen 3 × 2 képpontot tud megjeleníteni. Egyúttal fénytoll kezelésére is alkalmas. A jelenleg gyártott készülékekben fekete-fehér tévéképcsövet alkalmaznak megjelenítőként. A megjelenítés minősége kielégítő, jóllehet sokkal előnyösebb lett volna a szemet kevésbé igénybe vevő, kifejezetten számítástechnikai célokra szolgáló zöld vagy ámbrasárga, tükrözésmentes képcső beépítése.

A billentyűzetvezérlőhöz (TAST) TÁKI gyártmányú, Hall-generátoros klaviatúra csatlakozik. Ez jó használati minőséget biztosít, bizonyos megoldásaival azonban már kevésbé lehetünk elégedettek. Kényelmetlen, hogy „shift lock" állapotban a nem alfabetikus billentyűk is a felső állásnak megfelelő írásjeleket állítják elő, ami miatt vissza kell térni az eredeti állapothoz, ha a nagybetűk mellett az alsó állásnak megfelelő speciális jeleket kívánjuk elérni. A különböző irányú kurzormozgató billentyűket is szerencsésebb lett volna nem szigorúan egymás felett elhelyezni. Előnye a Comput–80 klaviatúrának, hogy valamennyi magyar ékezetes karakterrel és külön numerikus billentyűzetcsoporttal rendelkezik.

A nyomtató csatlakoztatására a párhuzamos be/kimeneti illesztőegység (PIOZ) szolgál. Ez 4 db BSI szabvány szerinti csatornát tud kezelni. Nyomtatók csatlakoztatására használható még ezenkívül a két darab, V.24 szerinti, soros vonal kezelésére alkalmas SIOZ kártya, illetve az egy soros és két párhuzamos be/kimenet illesztésére alkalmas SIPIOZ kártya.

A gyártó nem határozta meg előre a rendszerekkel szállított nyomtató típusát, hanem a piacon kapható, igen széles mátrixnyomtató-típusválasztékból vállalja, hogy beszerzi a felhasználó által igényelt típust, és azt felkészíti az ékezetes nemzeti karakterek kezelésére. A soros és természetesen a párhuzamos vonalak is használhatók más be/kiviteli funkciókra is, többek között adatátviteli célokra.

Az egymunkahelyes, moduláris felépítésű rendszerben a speciális hardverlehetőségek igen széles további választéka áll még rendelkezésre. A színes grafikai alrendszer, a matematikai processzor (MATH), a vonalkódolvasó és a mágnesszalagos háttértároló-egység a leginkább említésre méltó. A kártyaszintű modularitás legfőbb előnye éppen az, hogy elvben a legspeciálisabb felhasználói igényeket is ki lehet elégíteni a megfelelő kártya-, illetve csatlakozó perifériális rendszer kidolgozásával, amennyiben az nem áll rendelkezésre.

Többmunkahelyes rendszer

A prospektusokban még nem többmunkahelyes rendszerként feltüntetett Comput–80 időközben többfelhasználós kivitelben is elkészült. Az ilyen rendszer központi gépe a Comput–80 család legnagyobb kiépítésű tagja, a Comput–80/70. Az alapvető különbség kártya szinten a központiegység-funkcióban van. A CPUZ típusjelű processzorkártyán nem található memória, hanem helyette egy kiegészítő 4 címbites címkiterjesztési funkciót alakítottak ki. Ezzel 16 db, egyenként 32 kbájtos memórialapot lehet megcímezni. Az így elvben megcímezhető 512 kbájtnyi memóriából egyidőben csak 64 kbájtot láthat az éppen futó program, és lapváltással lehet hozzáférni egy másik területhez. A memóriát természetesen külön kártyákkal kell kiépíteni.

A központi gép egy nagyobb méretű, gurítható Kontaset szekrényekben helyezkedik el. Ebben több műszerfiók is elfér, de talán még ennél is fontosabb, hogy a maximálisan 2 db Winchester-mágneslemez egység és a 8 hüvelykes hajlékonylemez-meghajtók is ide építhetők be. A Winchester-lemezkapacitás határa így már 80 Mbájt, ami igencsak komoly hardverkapacitást jelent. Ilyen konfigurációban már gyors és nagy kapacitású mágnesszalagos alrendszerről is gondoskodni kell. A Comprojectnél tett látogatásunk során láthattuk azt a start-stop üzemmódú digitális kazettát, amely az alkalmazott kazetta típusától függően 8–40 Mbájt összállomány fájlszintű mentését és visszatöltését teszi lehetővé.

A többmunkahelyes rendszer felhasználói gépe a család legkisebb tagja, a Comput–80/10. Ez egy 64 kbájtos, egykártyás alapgép, amely 2 db párhuzamos interfésszel, 1 db soros interfésszel és 2 db programozható valós idejű órával rendelkezik. A billentyűzettel egybeépített alapkészülék kisméretű és formatervezett kivitelű (lásd a címlapot), kívül van viszont a meglehetősen nagy és súlyos tápegység, ami nem mondható a legkedvezőbb egykártyás megoldásnak. A felhasználói gép növelt átviteli sebességű (50 kbit/s hasznos sebesség) soros vonal segítségével csatlakozik a központi géphez. Megjelenítőként a korábban említett tévékép-csöves, pontosabban: átalakított tévékészülékes megoldás áll rendelkezésre. Láttuk azonban már a 80/10-hez csatlakoztatható, zöld képcsöves számítástechnikai monitor első példányait, amelyek sokkal kedvezőbb megoldásnak tűnnek.

A 80/10 önálló személyi számítógéppé is konfigurálható, mivel egy kártyányi bővítés elfér az alapkészülékben, és így a szükséges hajlékonylemez-illesztés megoldható. A hajlékonylemez-meghajtókat ebben az esetben azonban szintén külön házban kell elhelyezni.

A többmunkahelyes rendszerben maximum 4 db felhasználói gépet lehet csatlakoztatni. Ez a korlátozás nem a hardver kiépíthetőség adott korlátai miatt, hanem az alkalmazott operációs rendszer megoldás és a felhasználói tevékenységek adminisztrációja miatt áll fenn.

A szoftverjellemzők

Az alapszoftver részei: ROM rezidens monitorprogram; CP/M 2.2-vel felülről kompatibilis operációs rendszer; másoló (TRANS), rendszerállapot-kezelő (STAT), nyomtatási üzem-

mód beállító (PRN), képernyő-orientált szövegszerkesztő (EDIT); makro assembler (MASS), kapcsolatszerkesztő és betöltő (RLDR), nyomkövető programrendszer (SUB), parancsköteg-feladó (SUB); fájl-összehasonlító (FILC), lemeztartalom-módosító (DISK MONITOR); BASIC értelmező program.

A Comput–80-on az operációs rendszer két változatát kellett megvizsgálnunk. Az egymunkahelyes rendszereken az egyfelhasználós DOSY rendszer, a többmunkahelyes rendszereken a többfelhasználós MOSY rendszer használható.

A MOSY-ban az ún. BIOS (Basic Input Output System) ugrótáblájából hiányoznak a lemezkezelés belépési pontjai, azaz a felhasználó nem tud közvetlen módon, hanem kénytelen az ún. BDOS-on (Basic Disc Operating System) keresztül mágneslemezt kezelni. Megjegyzésünkkel kapcsolatban a CP/M-et nem ismerőknek annyit kell tudniuk, hogy a BIOS az alsó szintű perifériakezelést valósítja meg, a BDOS pedig a mágneslemezek fájlszintű kezelését támogatja. Az általa realizált fájlkezelő rendszer meglehetősen egyszerű, és ezért keveset tud. A BIOS ugrótáblája a CP/M szabványosított eleme, éppen azért, hogy az igényesebb fájlkezelést kívánó programok a BIOS-en keresztül megkerülhessék a CP/M beépített fájlkezelőjét. Az ilyen programok értelemszerűen nem fognak futni a MOSY alatt. A MOSY esetében tehát nem teljes mértékben áll fenn a felülről való kompatibilitás. A DOSY-t kipróbálva nem találtunk olyan specifikumot, ami miatt a CP/M-mel való kompatibilitás kérdéses lenne.

Mindkét rendszerben pozitívan értékeltük azt a sajátosságot, hogy a hibajelzések magyarul íródnak ki a képernyőre. Egyfelhasználós üzemben csak egy korlátot észleltünk mindkét rendszerben. Bár mindkét konfigurációban 64 kbájtos a beépített RAM kapacitás, a TPA mérete kisebb, mint más 64 kbájtos CP/M gépeken. Így találtunk olyan programot (egy fejlett C fordítót), amit más CP/M gépen tudtunk futtatni, a Comput–80-on a nem elégséges tárméret miatt viszont nem. A gyártók közlése szerint ez csak egy rövid átmeneti időszakra igaz, mert a TPA területe hamarosan lényegesen megnő.

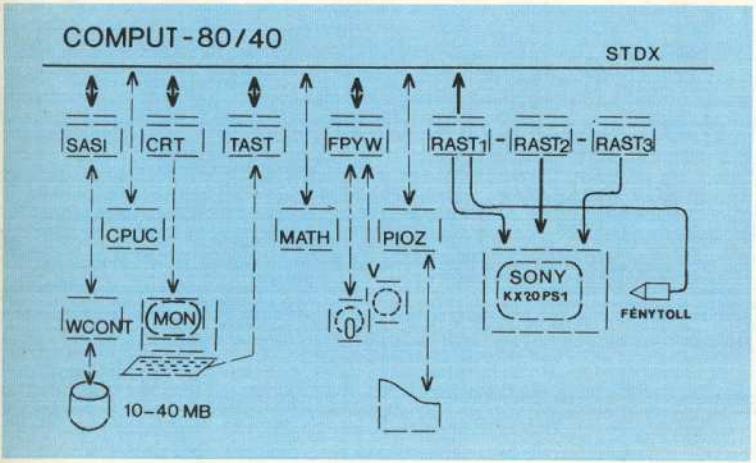
A többfelhasználós üzemmód támogatásának egyik eleme a CP/M 2.2-ben is meglévő ún. „user" parancson alapul. Ebben a parancsban meg lehet adni egy 0–15 közötti számot, amely kijelöli a futtatható programok és adatfájlok ún. felhasználói területét, amelyhez csak az adott felhasználó férhet hozzá. Hogy ne kelljen minden felhasználónak külön példányban tárolnia az általánosan használt dolgokat, de azért az eredeti CP/M „user" szám filozófia szerinti védelem is meglegyen, a MOSY konzolparancs feldolgozójából (CCP) a 0-ás területet minden „user" szám alól el lehet érni. Ez dicséretes megoldás, de nem ártott volna úgy továbblépni, hogy valami általános automatizmust vezetnek be, amikor például valamilyen bejelentkezési procedúra keretében automatikusan rendelődik egy-egy ilyen szám magukhoz a terminálokhoz. Így ugyanis olyan ütközéseket lehetett volna elkerülni, amelyek most könnyen előadódnak.

A fájlokhoz való konkurrens hozzáférések szabályozását három „general attribute" bevezetésével kívánja a MOSY támogatni. Az RWP attribútummal rendelkező fájlokat csak az a terminál olvashatja vagy írhatja, amelyik megnyitotta őket. A kizárólagos hozzáférést menet közben kiadott BLOCK művelettel lehet elérni. A SYS fájlokat mindenki olvashatja, de senki sem módosíthatja. Mivel az utóbbi kétállapotú fájl az eredeti CP/M-ben nem létezik, egy nem kifejezetten Comput–80-ra írott program csak RWP típusú fájlokat tud létrehozni. Ebből szintén adódhatnak bizonyos problémák.

A MOSY egyik funkcióját a gyártók spool mechanizmusnak nevezik, jóllehet csak sorba állított (maximum 10 elem) lemezes fájlok listázási funkciójának tekinthető. A spool mechanizmusnak ugyanis jóval nagyobb automatizmussal kell működnie és jóval nagyobb használati kényelmet kell biztosítania.

A MOSY által nyújtott többfelhasználós üzemmód teljesítőképességéről általában azt mondhatjuk, hogy a többfelhasználós rendszerben a felhasználók szinte alig éreznek valamit a hardverkiépítésből adódó nagy kapacitásból. A tapasztalatok szerint a fájlfeldolgozás hatékonysága nem számottevően haladja meg egy jóval lassúbb, hajlékonylemezes és egymunkahelyes rendszer teljesítményét. Ebből az is látszik, hogy igazi többfelhasználós rendszernek nem tekinthető a többmunkahelyes Comput–80/MOSY kombináció. Ezért feltétlenül szükségesnek tartjuk a MOSY megfelelő továbbfejlesztését.

Az egyéb szoftverelemeket csak felsorolásszerűen említjük meg, tekintettel arra, hogy ezek jól ismert CP/M szoftverek, amelyekről ezért nincs különösebb értelme részletes ismertetést vagy véleményt írni. Egy részük fordítóprogram (PASCAL, FORTRAN, C, COBOL, PL/1), más részük általános alkalmazás, mint az adatbázis-kezelő (DB) és a szövegfeldolgozó (Text). Rendelkezésre állnak még a hazai igényeknek megfelelően kifejlesztett célszoftverek is.



Kereskedelem és vevőszolgálat

A Comput–80-ra a következő irányárakat kaptuk a gyártóktól:

– Egymunkahelyes rendszer 512 kbájt hajlékonylemez-kapacitással és mátrixnyomtatóval: 420–450 eFt
– Többmunkahelyes rendszer 256, ill. 512 kbájtos memóriájú, 10, ill. 20 Mbájt Winchesteres központi géppel, az egyéb konfigurációs elemektől is függően: 900–1300 eFt
– Komplett DOSY alapszoftver: 140 eFt
– Komplett MOSY alapszoftver: 140 eFt + felhasználói munkahelyenként 40 eFt
– Fordítóprogramok: 60 eFt
– Adatbázis-kezelő: 60 eFt
– Szövegfeldolgozó: 50 eFt

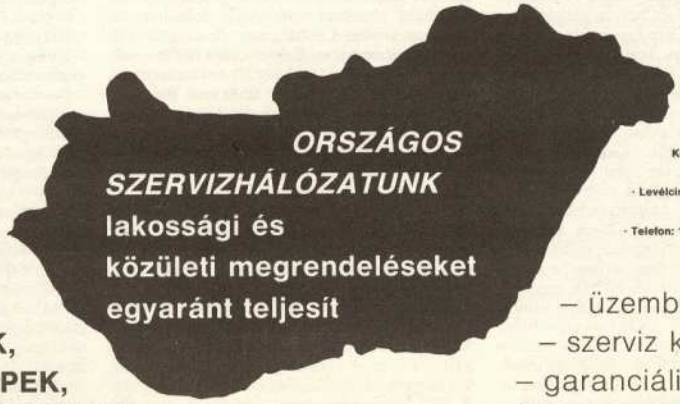
A berendezésekre 12 havi garanciát vállalnak. A szállítási határidő a megrendeléstől számított 6 hónapon belül van. Szerviz alapszolgáltatásként vállalják a hiba elhárítását 24 órán belül.

Összefoglalva azt mondhatjuk, hogy a hazai piacon kapható gépek közül a Comput–80 gépcsalád széles körű konfigurálhatóságával tűnik ki. Mivel tipikus személyiszámítógép-konfigurációban ára megfelel a hasonló gyártmányokénak, megítélésünk szerint elsősorban ott érdemes alkalmazni, ahol valóban szükség van a modulárisan bővíthető személyi számítógépre, illetve a többgépessé összekapcsolt rendszerkiépítésre egy kezdeti személyi számítógépes alapról.

NACSA SÁNDOR–SIMON IVÁN

SZOLGÁLTATÁSUNK ÖNRE VÁR

BENNÜNK SZERVIZ PARTNERT TALÁL!



VÁLLALJUK:

ELEKTRONIKUS
ÍRÓGÉPEK,
SZÁMOLÓGÉPEK,
PÉNZTÁRGÉPEK,
KISSZÁMÍTÓGÉPEK,
GYORSMÁSOLÓGÉPEK,
SZEMÉLYI SZÁMÍTÓGÉPEK,
SZÁMÍTÁSTECHNIKAI RENDSZEREK

ORSZÁGOS SZERVIZHÁLÓZATUNK
lakossági és közületi megrendeléseket egyaránt teljesít

– üzembe helyezését,
– szerviz kiszolgálását
– garanciális
– és –
– garanciaidőn túli javítását

Központ: Budapest V., Bécsi u. 8.
· Levélcím: 1368 Budapest, Postafiók 314
· Telefon: 184-899 · Telex: 22-4381, 22-6841

A számítógép-hobbizmus úttörői

Az első mikroszámítógépes hobbisták semmiképpen sem hasonlíthatók a mai hobbiszámítógépek amatőrjeihez. Ezek az emberek ugyanis komoly elektronikai és számítástechnikai felkészültséggel rendelkeztek. Erre már azért is szükség volt, mert a gépek nem voltak túlságosan megbízhatóak, a gyártótól igénybe vehető hardver- és szoftvertámogatás pedig nemcsak hogy szűkös volt, hanem a magánszemélyeknek meglehetősen drága is.

Az első hobbistákra ezért jellemző volt, hogy saját berkeiken belül meg tudták oldani problémáikat, sőt minden bizonnyal éppen ezek a problémák, az új technikai lehetőségek kihívása vonzotta őket mágnesként a mikroszámítógéphez. Legjobbjaikat azóta is a „hacker" jelzővel illetik, tükrözve azt, hogy az illető szinte átverekszi magát a megoldhatatlan problémákon, és *nem* szokványos megközelítési módjával igen ügyes és elegáns mérnöki megoldáshoz jut. A hackerek igen nagy hatással voltak a sikeres mikroszámítógép-konstrukciók kidolgozására. Elég, ha az Apple II-re vagy a Macintoshra hivatkozunk, mint két hírneves hacker, Steve Wozniak és Bill Atkinson munkájának alapvető eredményeire.

A hackerség gyökereit a 60-as évek egyes egyetemi kutatásainál (például Massachusetts Institute of Technology) és az interurbán távhívó rendszer belső működését feltáró és ingyen telefonálást támogató készülékek fejlesztésénél találhatjuk meg. Az utóbbi miatt vannak a „mozgalomnak" olyan külső bírálói, akik mint a jogellenestől sem visszariadó jelenséget, elítélik a hackerek tevékenységét. Az ilyen bírálat a hacker jelenség teljes félreismeréséből ered. A hacker célja ugyanis nem a törvény által megszabott korlátokkal való szembeszállás volt ebben az esetben sem, hanem egy bonyolult működésű és funkcionalitású rendszer megismerése és emberi „legyőzése", vagyis az egyén örök vágya, hogy diadalmaskodjék a természet felett.

A hackerek már szinte a kezdetek óta (lassan már vagy húsz éve) kisebb közösségekben tömörülve dolgoznak. A hackernek ugyanis szüksége van a hasonszőrűek környezetére, ahol megfelelően értékelni tudják teljesítményét, és ez további „hódításokra" serkenti őt. Ilyen közösségek voltak a 70-es évek közepének első számítógépes klubjai, és ilyenek ma a leginkább innovatív mikroszámítógépes vállalatok fejlesztő teamjei. A Macintosh fejlesztői például szinte egytől egyig hackerek voltak a javából.

1984 novemberében már országos konferenciájukra is sor került. Az erről szóló beszámolók igen izgalmasak (lásd a Byte 1985. márciusi számát), mi most mégis inkább azt idéznénk fel, hogyan is vallott a régi ősidőkben tevékenységéről a mozgalom nagy öregje, Captain Crunch (Ron Rosenbaum, Secrets of the Little Blue Box, Esquire, October 1971., ill. ugyanez The First Computer Freaks címmel az Esquire 1983. júniusi, jubileumi számában):

„Egy és csakis egy oka van annak, amiért ezt csinálom. Tanulmányozok egy rendszert. A telefonvállalat egy Rendszer. A számítógép egy Rendszer. Érti? Ha valamiért is azt teszem, amit teszek, az csakis az, hogy feltárjak és felfedezzek egy Rendszert. Számítógépek. Rendszerek. Ez az, ami engem különösen érdekel. A telefonvállalat semmi más, mindössze egy számítógép."

Ki is ez a Captain Crunch? Egyik, magát Gilbertsonnak nevező kollégája így mutatta be őt az írónak:

„Ő, a Kapitány! Ő valószínűleg a legnevezetesebb telefonőrült. Ropogtatás Kapitánynak nevezteti magát a hírhedt Cap'n Crunch 2600-as sip után. Évekkel ezelőtt ugyanis a Cap'n Crunch reggelire való gabonapehely-készítmény gyártói minden egyes dobozba egy játéksípot helyeztek el ajándék gyanánt. Valahogyan az egyik telefonőrült felfedezte, hogy a játéksip merő véletlenségből pontosan azt a 2600 ciklus/másodperc jellemzőjű hangszínt állítja elő, amit a távolsági telefonrendszer is használ bizonyos belső célokra. Amikor Ropogtatás Kapitányt a tengeren túlra, Angliába vezényelték katonai egységével együtt, temérdek telefonhívást kapott barátaitól. Ezeket »tompította«, vagyis ingyenessé tette őket úgy, hogy saját oldalán megfújta a Cap'n Crunch sípot.

Ropogtatás Kapitány az idősebb telefonőrültek egyike. Mérnök, akinek egyszer már baja származott a telefonnal kapcsolatos ügyeiből, de nem képes leállítani magát. Nos, ez a tag beutazza az országot egy Volkswagen mikrobuszban, amelynek hátuljába egy egész telefonközpont és egy számítógépesített, szuperfejlett multifrekvenciás készülék van beépítve. Egy elhagyatott autópályaszakaszon odaáll egy telefonfülkéhez, kisbuszából kivezet egy kábelt, ráakasztja a telefonra, és csak ül órákon, néha napokon keresztül a buszban, küldi egymás után a hívásokat az ország egyik végéből a másikba, a világ egyik végéből a másikba."

Ropogtatás Kapitány különösen büszke berendezésére. Így kérkedik az írónak:

„Utaltak a többiek a – hogyan is nevezzem – az én berendezésemre? Mit is mondtak? Csak úgy, kíváncsiságból kérdem: elmondták-e Önnek, hogy ez egy igen fejlett, számítógép-vezérelt készülék, a kimeneteket akusztikus csatolással fogadja, és olyan központja van, amely több vonalat is össze tud kapcsolni egymással? Mondták-e Önnek, hogy frekvenciatűrése garantáltan jobb, mint 0,05 százalék? Hogy az amplitúdótűrés kevesebb, mint 0,01 decibel? Azok az impulzusok, amelyeket Ön hallott, tökéletesek voltak. Mindössze gyorsabban érkeznek meg, mint a telefonvállalatéi. Azok az impulzusok nagy pontosságú műveleti erősítőkből származtak. A műveleti erősítőket a műszerekben való használatra tervezték, hogy ultrastabil erősítésük, szuperalacsony torzításuk és pontos frekvencia-jelleggörbéjük legyen. Elmondták-e Önnek, hogy berendezésem −55 °C és +125 °C közötti hőmérséklet-tartományban képes működni?"

A saját maga által épített szuperberendezésen túl a Kapitány legalább olyan büszke volt eredeti kísérleteire:

„Elmondták Önnek, hogyan hajtottam végre egy világ körüli hívást? Elmondom hát. Berendezésemmel titokban felhívtam Tokiót, az kapcsolt engem Indiába, ahonnan Görögországba kapcsoltak, onnan a dél-afrikai Pretóriába, majd Dél-Amerikába. Innen Londonba „mentem", egy londoni telefonkezelővel egy New York-ihoz kapcsoltattam magam, majd egy kaliforniai kezelőhöz, aki a mellettem levő telefont hívta fel. Felesleges mondanom, hogy kiabálnom kellett ahhoz, hogy halljam saját magamat. De a visszhang kinn volt a javából. Fantasztikus. Késleltetett. Húsz másodperccel volt késleltetve, de azért hallhattam magamat, amint saját magamhoz beszélek."

Ropogtatás Kapitányt és 15 évvel ezelőtti telefonőrültségét (a freak vagy phreak fogalmat magyar értelemben leginkább az őrült fejezi ki, amin persze szakmai fanatizmust kell érteni) azért volt érdemes felidézni, mert a lehető legjobban érzékelteti a hackerség lényegét. A telefonos hackerség továbbá azért is jó példa, mert a telefonrendszer a köznapi olvasó számára sokkal közérthetőbb környezet, mint a számítógépes hardver-szoftver bonyolult belső világa, amiben a mai hackerek oly elöszeretettel tevékenykednek. A rokon mindkettőben az, hogy a technikai világ jobb megismerése és a még nem kipróbált technikai lehetőségek feltárása a motivációs tényező. A másik cél, hogy minél többet kihozzanak a „rendszerből". Mindezek mögött pedig az bújik meg, amit a hacker konferencián Steve Wozniak így fogalmazott meg: „A hacker-mozgalom a bennünk rejlő gyermekiességet képviseli".

Ami magát a Kapitányt illeti, ő változatlan ifjonti hévvel hackerkedik tovább, bizonyítva, hogy a jelenkor számítógép-bűvölő gyerekeinél mennyivel többet értek a közelmúlt nagy öregjei. A Kapitány ilyen újabbkori „dobása" volt, hogy megmutatta, hogyan kell FORTH-ban igazi word processort írni (mert egy hacker számára ahhoz kétség sem férhet, hogy lehet). Terméke EasyWriter néven az IBM PC első ilyen eszköze lett 1981-ben, mi több, még a µMagazin olvasói is olvashattak róla az 1984. 4. szám 3. oldalán. Ma már felvett név mögé sem kell rejtőznie. Tudjuk, hogy becsületes nevén John Drapernek hívják. De mennyivel jobban hangzik, hogy Ropogtatás Kapitány!

(Nino)

A technológiák szerepe

A mikroszámítógépes piac megteremtése és a mikroszámítógép, mint egyéni használatra szolgáló, innovatív számítógépes eszköz megalkotása, egyértelműen Ed Roberts és az általa vezetett MITS cég érdeme. A néhány lényegi momentumhoz szintén hozzájáruló, de a MITS-et csak követő IMSAI cég érdeme másodlagos. 1978-ban azonban már egyiket sem találjuk a piacon. Roberts 1977 májusában dobta be a törülközőt. Megelégelte a hirtelen növekedéssel járó, számára egyre elviselhetetlenebb vállalatvezetési feladatokat, és cégét eladta a Pertec-nek. Az IMSAI sem tudott megbirkózni saját növekedésével, és 1978-ban kénytelen volt csődöt jelenteni.

A két cég után igen értékes örökség maradt. A társadalmi méretűvé vált számítógépes hobbizmus, a mikroszámítógépes lavinát továbbgördítő tényezőként, megteremtette a háziipari rendszerű mikroszámítógépes ipart. A hobbizmus krémjét alkotó, számítógép-építő megszállottak személyében létrejött az a kutatási-fejlesztési bázis, ami az új iparág még gyorsabb növekedéséhez szükséges volt.

Az olcsó és szabványos alapszoftverek piacát megteremtő Microsoft és Digital Research a két cég közvetlen leszármazottai. Egy sor más vállalat gyökerei is a MITS-hez és az IMSAI-hoz vezethetők vissza. Itt csak a kereskedelmi vállalkozásokat (The Computer Store, The Byte Shops, Lifeboat Associates és Computerland) és egy jelentős kiadási vállalkozást, a Personal Computing magazint (amelyből a PC, PC World és a Macworld is kinőtt a későbbiekben) említjük meg. A közvetett leszármazottak közül az Apple a legjelentősebb.

Apple-kezdetek

A kaliforniai HCC klub első találkozóira szinte véletlenségből keveredett Steve Wozniak, fiatal kora ellenére, már igen komoly „hacker" múlttal rendelkezett. Már kamasz korában televíziós áramköröket tanulmányozott, papíron vagy 50 számítógépet tervezett, oszcilloszkóppal írásjelek megjelenítésére alkalmas készüléket alakított ki, nemegy fordítóprogramot készített – csak úgy, hobbiból. Három évig az egyik legjobb kaliforniai egyetem számítógéptudományi szakán tanult, majd néhány évig az elektronikai szakma egyik „gyöngyszeménél", a Hewlett Packard-nál dolgozott chip-tervezőként. Ilyen alapos és mindenre kiterjedő szakmai múlt állt mögötte, mire 1976-ban megtervezte a megoldásaiban még ma sem meghaladott Apple II konstrukcióit. Közben volt azonban még az Apple I, ami mind konstrukciós, mind üzleti értelemben előfutára volt a későbbi sikernek.

Amíg HCC-s klubtársai az Altair alapokon építettek különböző dolgokat, addig Wozniak olyan kompakt és egyszerű számítógépet akart, amelyikkel otthoni tévékészüléke segítségével is kedvére szórakozhat. A leendő gép BASIC-jének megtervezése után hamarosan kikötött az akkor kapható legolcsóbb, 6502 típusú mikroprocesszor mellett. Mivel igencsak kevés pénze volt, a szükséges funkciók és az alkalmazott chipek „összepasszításával" egy mindössze 30–40 chipből álló, egykártyás gépet tervezett. A gép 4 kbájtos memóriájába betöltött BASIC rendszer a házi tévékészüléket írógépszerű kiíratásra használta, és így tökéletesen alkalmas volt arra, hogy BASIC programokat lehessen önállóan írni és futtatni rajta.

Wozniak barátja, az övéhez ugyan nem hasonlítható, de szintén alapos mérnöki felkészültséggel rendelkező Steve Jobs észrevette, hogy klubtársaik jó része szívesen meg is vásárolna egy ilyen gépet. Jobs végzett néhány alapvető kalkulációt, sőt még egy 100 darabos megrendelést is felhajtott a helyi The Byte Shops-ban, majd rábeszélte Wozniakot, hogy tevékenységüket helyezzék üzleti alapokra.

Jobs eladta mikrobuszát, Wozniak a HP kalkulátorát, az alkatrészeket pedig 30 napos hitelre sikerült biztosítani, így aztán belevághattak vállalkozásukba. A 666 dollárért forgalomba hozott Apple I már 8 kbájt memóriával rendelkezett, és csak a megfelelő billentyűzetről, transzformátorról és a video-megjelenítőhöz való csatlakozásról kellett gondoskodnia a vevőnek. A gépből Jobs és Wozniak kb. 10 hónap alatt 200 darabot gyártott munka után, egyikük garázsában.

Az Apple II születése

Az Apple I megjelenése sajátos piaci és konstrukciós felülvizsgálatot indított el a klubban. Ekkoriban jelent meg a Cromemco színes grafikai Dazzler kártyája az Altairhez, és egy színes grafikai megjelenítésre képes miniszámítógépben is gyönyörködhettek a klubtagok. A kihívást az jelentette Wozniaknak, hogy a lehető legkevesebb számú, közönséges katalógus áramkör felhasználásával lehet-e olyan új konstrukciót tervezni, amely az otthoni színes tévékészüléken színes grafikát is meg tud jeleníteni. Először egy 40 × 48 képpontos üzemmódot, majd néhány további chip alkalmazásával és ügyes trükkökkel 280 × 192 képpontos, nagy felbontású grafikát sikerült elérnie.

Mivel akkoriban még nem voltak speciális grafikai chipek, Wozniak egy azóta is egyedülálló villamosmérnöki és rendszertervezői csúcsteljesítményt nyújtott. Más vonatkozásban is minőségi előrelépést hozott az Apple II. A 16 kbites memória-áramkörökkel 16 kbájt memóriát lehetett olcsón kialakítani. Az alapkártyán (ún. motherboard) elhelyezett igen egyszerű és hatékony bővítési rendszer 8 kiegészítő kártyát volt képes befogadni (mindössze 24 érintkezős csatlakozók). Az így már igen nagy teljesítményt nyújtó, moduláris továbbépíthetőséget támogató alapkártyát mindjárt az első, 1000 darabos sorozatnál igen olcsón, mindössze 250 dollárért elő lehetett állítani. Nem kevesebb, mint hét évig lehetett így ez az alapkonstrukció az Apple vihargyors növekedésének egyedüli alapja.

Az új minőség azonban az új lehetőségek oldaláról jelentkezett elsősorban. A Wozniak által beprogramozott első alkalmazás az 1975-ben megjelent, első igazi videojáték, a Breakout (kitörés) programozott szimulációja volt. Ma úgy mondanánk, hogy játékprogram, akkor viszont ilyesmi még nem létezett, hiszen a videojátékokat célhardverrel valósították meg. Ennél a kifejezetten csak videóban megvalósítható ügyességi játéknál egy többrétegű téglafalhoz kell ütni a labdát. Az érintkezés hatására az adott tégla eltűnik, és a labda sebessége megváltozik. A játékos megfelelő segédeszközzel befolyásolja közvetlen módon a képernyőn történteket, és természetesen folyamatosan észlelnie kell a változásokat (közvetlen manipuláció, vizuális visszacsatolással).

Az Apple II-t a Breakout alkalmazással mutatta be Wozniak a klubban. Ma így emlékszik erre vissza: „Ez volt életem legboldogabb napja. Óriási előrelépésnek tűnt nekem mindez. Hardver videojátékok korábbi tervezőjeként *tudtam*, hogy BASIC-ben való programozhatóságuk megváltoztatja majd az egész világot."

Ma már tudjuk, hogy nem is elsősorban a játék típusú alkalmazásokra lett igaz ez a világmegváltás. A közvetlen manipuláció és a vizuális visszacsatolás, mint a számítógéppel támogatott feladatmegoldás új technikai lehetősége, azzal a sajátossággal ruházta fel a személyi számítógépeket, amely az univerzális, programozható gépek között egyedül rájuk jellemző. Helyet biztosított tehát nekik a nap alatt, a lavina egy újabb hulláma indulhatott el ezzel. Vevőnél 1977 májusában helyezték üzembe az első Apple II-t. Két év múlva két olyan termék is megjelent az Apple II-re, ami a számítógép-alkalmazásban új technológiát honosított meg.

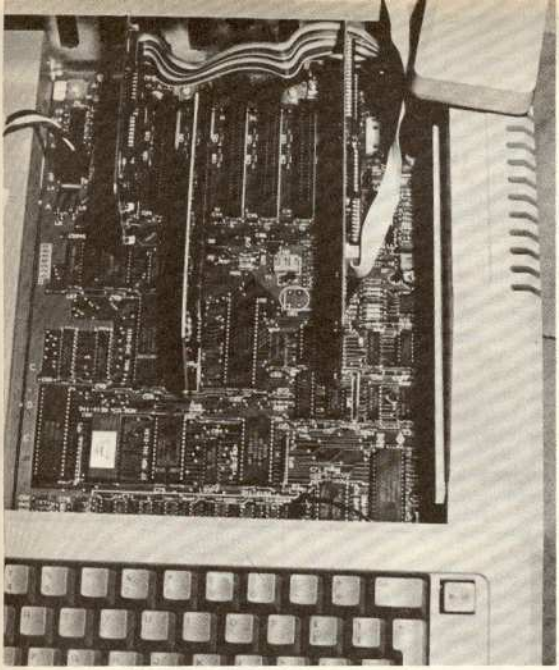
Képernyős szófeldolgozás

A szöveges dokumentumok képernyős kezelésének és feldolgozásának technikáját 1974-ben dolgozták ki egy terminálokkal ellátott Wang miniszámítógépre. Ennek elvi tervezője a Georgia állambeli Talmadge szenátor ügyvivő titkára, John Hayes volt. Amikor a szenátort kineveznék a Nixon elnök visszaéléseit vizsgáló Watergate bizottságba, levelezése oly mértékben megnőtt, hogy Hayes nem látott más kiutat, csak a megfelelő számítógépes feldolgozást. A profi irodai dolgozók számára kidolgozott rendszer nemcsak arról győzte meg Hayest, hogy a számítógéppel támogatott megoldással hatékonyabb és rugalmasabb az irodai munka gépesítése, hanem azt is bizonyította, hogy mindez interaktív kisszámítógépes megoldással a leggazdaságosabb.

Az Altair hirdetésre idejekorán felfigyelve, más üzletemberekkel együtt felkereste a MITS-et, és 1976-ban elindították az Altair Software Distribution Companynek nevezett közös vállalkozást. A Wang által szóprocesszornak nevezett rendszer mikrogépesítésétől ebben az időben még el kellett tekinteniük, mivel akkor még nem voltak a mikrogépekhez hasonlóan *olcsó*, profi igényeket (80 oszlopos megjelenítés, numerikus billentyűzet, szemet nyugtató, fekete alapon zöld tónusú képcső) kielégítő képernyős terminálok. Az egyént kiszolgáló alkalmazással szemben így inkább a szervezetek általános üzletvitelének mikrogépes támogatására koncentráltak.

1978-ban Peachtree Software néven önállósodott cégük uralta az általános üzletviteli célfunkciók, mint a könyvtartozások és -követelések nyilvántartása, megrendelésfelvétel és számlázás, bérelszámolás, a főkönyv vezetése stb. piacát.

Mire 1981-ben az akkori legnagyobb független szoftvergyártó, az MSA felvásárolta őket, már 25 ezret eladtak a mai mikrogépes



Egy Apple II belülről. A legfontosabb funkciókat ellátó alapkártyán kialakított csatlakozókba lehet bedugaszolni a különböző kiegészítő funkciókat ellátó bővítő kártyákat. A konstrukciós megoldást az IBM is átvette saját gépeiben.

szemmel egyáltalán nem olcsó, 1000 dollárnál drágább programcsomagjaikból.

Mindezek egyértelműen megmagyarázzák, hogy hiába készítette el valaki más – Michael Shrayer – az első mikroszámítógépes szófeldolgozó programot (Electric Pencil, eredetileg az Altair gépre), hiába adaptálta azt nem kis munkával az újabb mikrogépekre, fáradságos munkássága akkor még nem hozhatta meg a kívánt üzleti sikert. A képernyő-orientált, igazi személyi számítógépnek kellett megjelennie ahhoz, hogy a számítógépes szófeldolgozás meghóditsa a világot.

A tévékészülékhez való csatlakoztatásra tervezett Apple II csak 40 oszlop szöveg megjelenítésére volt képes, és a kisbetűs megjelenítéshez külön kiegészítő szoftverre volt szükség. Tökéletesen alkalmas volt viszont az alkalmi gépelők és a dokumentumkészítők számára, akik történetesen más alkalmazásra is használni kívánták gépüket. A számukra készült Apple Writer szófeldolgozó szoftvert 1979-ben hozta piacra az Apple cég. Az elsajátítás roppant könnyen ment, mivel maga a szoftver magyarázta meg a képernyőn, hogy hogyan is kell használni.

A képernyős szófeldolgozás még alig volt 5 éves, amikor ezt a még profi leíróknak és dokumentumkészítőknek is nagyon új technikát a széles publikum már tömegesen birtokba vehette. Az olcsó ár (valamivel 100 dollár felett) és az általános használhatóság új piaci szituációt eredményezett. Ugrásszerűen nőttek az eladások, és hamarosan 1 millió dollár jogdíjat fizethetett ki a szerzőnek az Apple. Az Apple Writerrel egyidőben azonban egy még nála is sokkal sikeresebb üzleti-irodai szoftver jelent meg.

Vizuális kalkulátor

Az Apple II és a vele egyidőben megjelent többi személyes „munkaeszköz" (Commodore PET, Tandy TRS–80) arra indította Dan Fylstrat, hogy szoftverkiadó vállalatot alapítson. Ekkoriban még nem volt ezekhez a gépekhez floppy lemez, csak kazettás magnó, így a Personal Software nevet viselő új cégnek játékprogramokkal kellett kezdenie tevékenységét. Első, egymillió dolláros bevételüket az Apple II-re kidolgozott Micro Chess program hozta. Több, mint 50 ezer példányt adtak el az Apple II képernyőjét sakktáblává alakító ügyes játékból. Az apró cégnek 1978-ban már elég tőkéje volt ahhoz, hogy anyagi bázist teremtsen a személyi számítógépes szoftver forradalmát elindító VisiCalc fejlesztéséhez.

Fylstra ekkor még a Harvard Business School hallgatója volt. A vállalati pénzügyek tanára hívta fel figyelmét egyik évfolyamtársa, Dan Bricklin elképzelésére, aki „vizuális kalkulátor" program készítésén gondolkodott. Fylstra adott Bricklinnek egy Apple II-t, amelyen elkészíthette a program első változatát. Ez meggyőzte Fylstrat, hogy az elképzelés nagy üzleti siker lehet, forgalmazási szerződést kötöttek, és a Personal Software anyagi támogatásával megkezdődhetett az igazi fejlesztés.

Bricklint az indította a VisiCalc fejlesztésére, hogy meglehetősen unta a vállalati pénzügyi tervek készítésének időrabló és állandó javításokat igénylő módszerét, amivel tanárai nyaggatták. Oszlopokból és sorokból álló, számszerű összefüggéseket tartalmazó, nagyméretű táblázatokat kellett több menetben, kézzel kezelni és átszámolni ahhoz, hogy eljussanak az optimális tervhez.

A nagyméretű, rovatokra osztott lapot, az ún. spreadsheet-et célszerűnek látta a személyi számítógép memóriájában kialakítani. A képernyőt gyorsan mozgatható ablakká alakítva, a táblázat egy részét közvetlenül manipulálhatóvá akarta tenni (például adatok bevitele vagy módosítása). A táblázat egyes rekeszei között fennálló függvényszerű összefüggéseket (például az adott oszlopban szereplő értékek összege) szintén a táblázathoz tartozóan, a gép memóriájában kívánta tárolni. Így a megfelelő adatok bevitele után azonnal megjelenik az eredmény, és ha nem kielégítő, akkor a megfelelő kiindulási adatok közvetlen átszerkesztésével mindjárt módosulnak a vonatkozó eredményadatok is.

A képernyőn látható, közvetlenül definiálható és manipulálható kalkulátor számítógépes megvalósítása közelebb állt Bricklinhez, mint maga a pénzügyi téma. A vezetői szakra azután került, hogy villamosmérnöki és számítógéptudományi szakképesítést szerzett, ezt követően pedig a vezető minikomputer-gyártónál (DEC) dolgozott egy számítógépes fényszedő, illetve minigépes szófeldolgozó rendszer fejlesztésén. Termékét képernyős számprocesszornak vagy számtáblázat-feldolgozónak is lehetne nevezni, ő azonban a látható kalkulátor (visible calculator) nevet választotta. A mai köztudatban electronic spreadsheet vagy electronic worksheet néven szerepel inkább, abból kiindulva, amit imitál.

A számtáblázatos adatok képernyős/számítógépes kezelésének és feldolgozásának technológiája született meg így Bricklin elképzelése nyomán. A hatékony megvalósításhoz igazi programozó „hacker" kellett, mivel ezt aligha lehetett interpretatív BASIC-ben termékké formálni (mellesleg a korábban említett szövegfeldolgozókat sem). Bricklin barátja, Bob Frankston pontosan ilyen adottságú ember volt. Könnyen kezelhető dokumentációt kellett még készíteni, amit Fylstra kitűnően meg is írt. Így az egyéves komplex fejlesztőmunka nyomán nemcsak egy innovatív termék, hanem minőségileg is kiválóan kivitelezett szoftver kerülhetett 1979 októberében a piacra.

Alkalmazás új módon

Már az első hónapban tapasztalták a fejlesztők, hogy igen olcsó, mindössze 150 dolláros programjuk az első igazi szoftver-sláger lesz. 9 hónapon belül adták el az első 10 ezer példányt. A vevők között vállalati elnökök, gazdasági igazgatók, beruházási bankárok, könyvelők, ügyvédek, orvosok, farmerek és sokan mások voltak. Nem meglepő, hiszen nincs olyan tisztán vagy részben szellemi foglalkozás a világon, melynek során fejlett piaci viszonyok között és éles gazdasági versenyben ne kellene szinte lépten-nyomon számtáblázatok feldolgozásával foglalkozni.

Bricklin így látta az első év után a VisiCalc alkalmazási tapasztalatait: „Sokan használják a VisiCalc-ot az eladások, az üzleti haszon, a részvénytőke-értéknövekedés és egyéb dolgok előrejelzésére. Mások az üzleti vállalkozás tervének készítésénél alkalmazzák, kiaknázva „mi van akkor, ha" képességeit. Még mások általános könyvelési, egyszerű leltárkezelési és fizetési lista készítésére használják. Vannak olyan mérnökök, akik optikai tervezésre, vagy például petrokemikáliákkal kapcsolatos dolgokra, sőt olyan orvosok, akik a munkájuk során adódó számításokra alkalmazzák. A legtöbb felhasználó azért szereti, mert könnyű a használata, és programozás nélküli alkalmazás-fejlesztést tesz lehetővé."

A VisiCalc elvű gépi támogatással a feldolgozások több *fajtáját*, az egyszerű összegző nyílvántartástól kezdve az igen összetett, magas szintű döntéshozatal támogatásáig meg lehetett fogalmazni. Tehát generikus alkalmazás volt. Ehhez programozók sem kellettek, vagyis alkalmazásgenerátor is volt. A feldolgozás során olyan közvetlen ember–gép kapcsolat, szimbiózis jött létre, amivel olyan döntéstámogatási rendszerek előtt is megnyílt az út, amelyeket még a legnagyobb számítógépeken sem lehetett addig, tisztán praktikus okokból, megvalósítani. A problémával teljes mértékben tisztában levő felhasználó heurisztikus megoldóképessége a „mi van akkor ha" próbálkozásos módszerével menet közben segíthette a hardver szempontból kis teljesítményű személyi számítógépet.

A VisiCalc tehát az eredetileg

elképzeltnél is jelentősebb technológiai innovációnak bizonyult. Ráadásul egy addig még alig feltárt területen, az üzleti/professzionális személyiszámítógép-alkalmazási szférában. A mikrogépes lavina tovább erősödött. Ennek első hazonélvezője az Apple volt, mivel majd egy évig csak ezen a gépen lehetett VisiCalc-ot használni. Az Apple II ázsiója az üzleti/professzionális alkalmazásokban ugrásszerűen megnőtt. Vajon ki tudná megmondani, hogy a napjainkig eladott 850 ezer VisiCalc példány hány Apple és hány más gép eladását segítette ezen a fizetőképes területen?

Olcsó mikroperifériák, nagy tömegben

Az Apple II-nek nagy előnye volt a piacot megalapozó többi személyi számítógéppel szemben, hogy nyitott rendszer volt, amit bővíteni lehetett nemcsak memóriával, hanem a megfelelő minőségű mikroperifériákkal is.

A legkézenfekvőbb példa a képernyő. Képernyőként ugyanis nemcsak otthoni tévékészüléket, hanem a szemet nem fárasztó, jó minőségű videomonitort is lehetett használni. A 24 sor, 40 oszlopos szöveg üzemmódot is fel lehetett javítani egy kiegészítő kártyával 80 oszloposra. Így a 70-es évek végén, amikor még a legolcsóbb képernyős terminál is 1000 dollárba került, az ennél néhány száz dollárral drágább Apple II-vel nemcsak ezt a funkciót, hanem személyi számítógépet is kapott a vásárló.

A szófeldolgozó és VisiCalc funkciókhoz szükség volt a mikroszámítógéphez igazodó méretű, hajlékony mágneslemezes tárolókra is. Az Altair jellegű mikroszámítógépekhez használt, 8 hüvelyk méretű hajlékonylemezeket nehézkes volt a kompakt, asztali kivitelű személyi számítógépekkel együtt adni. A 8 hüvelykes hajlékonylemezek piacán vezető Shugart Associates előrelátóan, már 1976-ban piacra hozott egy kisebb méretű, 5 1/4 hüvelykes változatot.

Ezeket a floppykat nemcsak olcsóbban lehetett előállítani, hanem végre alkalmazni lehetett őket az egyre nagyobb tömegben gyártott adatgyűjtő terminálokban, programozható műszerekben, speciális szófeldolgozókban és kisszámítógép-rendszerekben – egyszerű, beépített módon. A piac egyre bővült újabb gyártókkal, az árak folyamatosan estek, így egy önmagát erősítő, saját lavina indult be a mágneslemezes, kisméretű háttértárolók piacán.

Különleges szerep jutott ebben a folyamatban az Indiában nevelkedett Jugi Tandonnak. Miután az USA-ban egyetemi végzettséget szerzett, az IBM-nél dolgozott a mágneses perifériák területén, majd a lemezgyártással is foglalkozó Pertec-nél, míg végül 1976-ban megalapította saját vállalatát. Akkor még csak a floppy-meghajtókhoz szükséges fejekkel foglalkozott. Az általa tervezett, igen egyszerű fej darabonként mindössze 20 dollárba került, míg a konkurrens gyártmány ára elérte a 100 dollárt.

1979-ben már a komplett meghajtók piacára is belépett. 5 1/4 hüvelykes egysége nagy tételben 225 dollárba került, szemben a konkurrencia 450 dolláros árával. A magyarázat: minden részegységet maga állított elő, és olcsó munkaerővel, elsősorban a Távol-Keleten dolgoztatott. 1983-ban már 200 ezer darab meghajtót gyártott havonta, a nagy tételű ár pedig 160 dollárra csökkent.

Az Apple II-re elsőnek jelent meg 5 1/4 hüvelykes floppymeghajtó a nagy tömegben gyártott, új személyi számítógépek között. Wozniak itt is kitűnő munkát végzett. Ismét speciális chip vagy bonyolult vezérlő kapcsolás nélkül, egyszerűen egy kiváló áramköri megoldással tudta megvalósítani maximálisan 6 db, egyenként 140 kbájtos (formátumozott) meghajtó csatolását. A meghajtó adatátviteli sebességét is maximálisan kihasználó csatolás így ismét a lehető legolcsóbban juttatta hozzá az Apple-t a lavina erősödésének egy újabb technológiai tényezőjéhez. Természetesen ő maga is hozzájárult ahhoz, hogy egyre több 5 1/4 hüvelykes floppy-meghajtóra legyen igény.

A mikroszámítógépes lavina ezen időszakában a külföldi gyártók közül egyedül a japánoknak, és nekik is csak az olcsó nyomtatók területén sikerült meghatározó szerephez jutniuk. A max. 120 karakter/mp sebességű, asztali kivitelű berendezésekből 278 ezer darabot adtak el 1980-ban az USA-ban. Az ilyen, 1000 dollárnál olcsóbb nyomtatók piacának több mint 50%-át kaparintották meg maguknak a japánok, amikor valósággal elárasztották termékeikkel a piacot. Közöttük is vezető pozícióhoz jutott az Epson. MX–80 típusjelű, 80 karakter/mp sebességű mátrixnyomtatóját 650 dollárért hozta kiskereskedelmi forgalomba, a gyártóknak pedig nagy tételben mindössze 300 dollárért kezdte el árusítani.

A hozzáértők szerint itt is sajátos technológiai tényezők játszottak szerepet. A kandzsi írásjeleket ugyanis csak finomabb felbontásban lehet mátrixpontos alapon képezni, számuk miatt pedig sokkal korábban szükség volt az ilyen megoldásra az irodai számítógépekben, így a japán mátrixnyomtató-gyártók technológiai előnyhöz jutottak. Az Epson esetében az 1964-es tokiói olimpia eredménynyomtató rendszeréig nyúlik vissza ez az előny. A többi már a japán alapossággal célszerűsített konstrukción és az ehhez igazodó, nagy termelékenységű gyártás szervezésén múlt.

N. S.

(Folytatjuk)

MEDITÁCIÓ

Különösen izgalmas hónapok ezek a mostaniak a világ számítástechnikai fejlődésében. A mikro-számítástechnika korszakos fordulópontjához érkeztünk.

A személyi számítógépek széles körű elterjedésével fémjelzett, első forradalmi szakaszban a mikroszámítógépek milliós és tízmilliós nagyságrendben hódították meg szinte minden területét az életnek. Megtalálhatók az otthonoktól kezdve a vállalatokig és intézményekig terjedően szinte mindenütt. A 70-es évek második felének útkereső időszakában még lenézően megmosolygott gépek kivívták a legarisztokratikusabb hajlamú, a korábbi technikával elkötelezett szakemberek tiszteletét is, ami nem kis eredmény a pozíciófeltó mai világban.

Merre gördül tovább a mikroszámítógépes lavina? Mit hoz a mikroszámítógépes forradalom második szakasza? Tulajdonképpen semmi meglepőt, csak azt, amit a tudományos, technikai és gazdasági tényezők ismeretében már három évvel ezelőtt is meg lehetett jósolni (Számítástechnika, 1982. március, 2. oldal és 1982. április, 2. oldal).

Az eddig jobbára elszigetelten működő mikroszámítógépes alkalmazások átalakulnak, továbbfejlődnek *hálózaton keresztül egyesített, osztott felépítésű informatikai rendszerekké*. Maga az eddig ismert személyi számítógép is átalakul *új elvű személyi számítógépes munkaállomássá*.

A közvetlen manipuláció és a „csak az van, amit lát" új elvét a Xerox cég vezette be 1981-ben a Star munkaállomás piacra hozatalakor. A Xerox Star technológiája és egy még piacra nem hozott fejlesztése, a Smalltalk ment át az Apple Lisa technológiába, majd tovább, a már tömegesen értékesített Macintosh-ba. Mi több, az új elv egyik váratlan melléktermeke, hogy sikerült magát önmaga segítségével bemutatnia a magazin „élettelen" újságoldalain (μM, 1985. 1. szám, 26., 27. oldal). Így már nálunk is igen sokaknak lehet érzékletes vagy kvázi érzékletes sejtése arról, hogy miről is van szó.

A Mac szinte csak primitív előfutára annak, ami hamarosan széles körben elterjed. Már az „ősanyának" tekinthető Star is minden szempontból felette állt a Macnek. (Furcsa anyai kapcsolat, nemde?) Manipulációs környezete kiterjed a hálózati és osztott adatbázis-alkalmazásokra is, emellett teljesen homogén, vagyis nincs például szükség az egyes részkörnyezetek (szövegfeldolgozás, rajzolás, adatbázisból leképzett tabló stb.) közötti, meglehetősen nehézkes, Cut/Paste típusú adatkommunikációs mechanizmusra. Az operációs rendszer is teljesen beleolvad a homogén környezetbe. Egyébként akit érdekel a dolog, olvassa el a Byte 1982. áprilisi számának 242. oldalán megjelent cikket.

Vegyük még hozzá mindazt, amit az elmúlt 4 év fejlesztései hozzátehettek a Star gondolathoz (vagy minőségileg revidiálták azt), és máris ott vagyunk az elkövetkező évek új elvű személyi számítógépes munkaállomásánál. Honnan várhatjuk mindezt? Természetesen a Xeroxtól, az IBM-től, az AT & T-től, a Wangtól – szóval a nagyoktól. És még egyéb helyekről is. Például ma már ismert, hogy a Commodore a fejlesztés alatt álló szupergéppel együtt felvásárolta az Amiga nevű céget. Mégis, mit várhatunk ahhoz képest, amit eddig érzékelhettünk? Fantasztikusabbnál fantasztikusabb dolgokat. A Xeroxtól például a hangfelismeréssel rendelkező munkaállomást. (Nem ötven-hatvan szó, hanem tízezer szavas szótár!) A titkoló-

dzás nagy, a meglepetések ennek megfelelően még nagyobbak lehetnek.

A munkaállomásokat magába foglaló, hálózaton keresztül egyesített, osztott felépítésű informatikai rendszer elvi felépítése sem fog különbözni attól, ami már évekkel ezelőtt várható volt. Az alapvető átviteltechnikai rendszerben a hagyományos telefonhálózatokhoz hasonló heterogén megoldásrendszer alkotja az ún. multihálózati alapot. Az egyes átviteltechnikai megoldások ma már szinte teljesen szabványosítottak, és a protokollok egy közbenső szintjén egységesítik a multihálózatot. Külön ki kell emelni az egyik legfontosabb

SZÁMÍTÓGÉPESÍTÉS ÉS ELEKTRONIZÁCIÓ

területre, a nem kapcsolással működő, helyi digitális hálózatok (local area network – LAN) kommunikációs rendszerére kidolgozott IEEE 802 szabványcsoportot.

A multihálózatra ráépülő hálózati rendszerek architektúrájában azok az alapelvek fognak érvényesülni, amelyeket az élenjáró Xerox rendszerben lehetett annak idején felfedezni (Számítástechnika, 1982. július–augusztus, 4., 9., 10. oldal). A bázisgépekből kialakított különféle célgépek (állománykiszolgáló, kommunikációs kiszolgáló, clearinghouse stb.) együttműködő rendszere adja azt a rendszerarchitektúrát, amibe a szintén célgépnek minősülő munkaállomások integrálódnak. A kérdés tulajdonképpen csak az, hogy menynyire fogják megtartani korábbi terminálhálózati architektúrájukat a hagyományos gyártók, és hogyan tudják beilleszteni elképzeléseiket a nyílt architektúra elkerülhetetlen koncepciójába.

A hálózati rendszerek széles körű alkalmazása az információ tárolásának, kezelésének és feldolgozásának új minőségéhez vezet. Az így létrejövő, hálózaton keresztül egyesített, osztott felépítésű informatikai rendszert úgy kell felfognunk, mint a korábbi merev, papír alapú rendszertől dinamikus jellegével alapvetően különböző, új közeget.

Egyfajta dinamizmus jellemző erre a közegre akkor, amikor az információt a munkaállomás megjeleníti, kezeléséhez és feldolgozásához különféle segédeszközöket ad (gondoljunk például a MacPaint-tel való rajzolásra, átrajzolásra stb.). Egy másik dinamizmus jellemző akkor, amikor a hálózati rendszerben bármilyen távolságra, tetszőleges számú munkaállomásnak és tetszőleges értelmezésben teszi hozzáférhetővé az információt. Hozzátehetjük még, hogy szinte pillanatok alatt. Végül egy harmadik dinamizmus jellemzi akkor a közeget, ha dinamikai tulajdonságait én magam szabom meg, vagyis saját magam is programozom a viselkedését. Itt persze semmiképpen sem a BASIC vagy más hagyományos programnyelv használatáról van szó, hanem az adatbázis definíciójáról, a döntési rendszer meghatározásáról, szóval olyasmikről, amiket az alkalmazásgenerátorok tesznek lehetővé.

A kérdés ezek után az, hogy miért is fontos nekünk mindez, amikor többszörösen el vagyunk zárva ettől a technikától, és már az eddigi számítógépesítésben is jócskán lemaradtunk.

Több, egymást kiegészítő válasz adható erre a kérdésre:

- Ezek az új termékek túlmutatnak a játékgéppé degradált, vagy egyszerű BASIC-esdire kárhoztatott mikroszámítógép-szemléleten, de még az eddigi személyi számítógépeken is.
- Azt helyezik előtérbe, ami az eddigiek során itthon rendre a háttérbe szorult, vagyis az *informatikát*.
- Az eddig kézlegyintéssel elintézhető, elméletinek tűnő fejtegetések helyett a maga valóságában meg fog előttünk jelenni a korszerű informatika új minősége, amire már a súlyos gazdasági és társadalmi kihívás miatt is kénytelenek leszünk odafigyelni.
- A hálózaton keresztül egyesített, osztott informatikai rendszer meghatározza az informatika legfejlettebb formáját, így az ún. logikai módszerrel való tanulmányozása az informatika lényegi feltárásához vezető legkönnyebb út.

Miért nem elégedhetünk meg az informatika mikroelektronikán vagy az elektronizáción belüli tárgyalásával, ahogy társadalmigazdasági programunk koncepciója teszi? (Műszaki Élet, 1985. január 19-i szám melléklete)

A fentiekből következőkön túlmenően ismét több okot sorolhatunk fel:

- Az informatika a gazdaságilag legfejlettebb országokban a munkaerő nagy részét foglalkoztató szektor (például az USA-ban több, mint 50%), így csak ebben a minőségben vizsgálható, és nem egy technikai innováció részeként.
- A tények azt mutatják, hogy a korszerű elektronika alkalmazásának messze legnagyobb területe az informatika, így inkább az határozza meg a korszerű elektronika fejlődését, semmint a fordítottja. (A mikroelektronika fejlődését a MOS memóriák iránti számítógépes igények hajtották; időközben melléktermékként megjelent a mikroprocesszor, az ebből lett

mikroszámítógép a nagy létszámú és döntő jelentőségű szektorban szinte korlátlan felvevő piacra lelt, ez ugrásszerűen növelte a MOS memóriák iránti igényt stb. (lásd a lapunk 1985/3. számában megkezdett elemző sorozatot).

- Az informatika vezető szektorrá válása a gazdasági fejlődéssel szoros összefüggésben, az utóbbi 40 év alatt bekövetkezett folyamat. A folyamat legfőbb mozgató ereje az volt, hogy a legkülönfélébb gazdálkodó egységek először a pénzeszközök, később a jól képzett és lojális munkaerő, majd az ezekre is ráépülő információ tudatos kezelésével egyre jobban elsajátították a hatékonyságnövelés ezen újabb erőforrásait.
- A fentiekből egyértelműen következik, hogy az elektronizációnál fontosabb a számítógépesítés oldaláról közelíteni a dolgokat, illetve az informatikai szektor újabb hatékonyságnövelési erőforrások tényleges bekapcsolásával elért súlyának oldaláról vizsgálni az elektronizáció jelenségeit.

Az elmondottakból következően az elektronizációs program csak akkor lehet jó, ha egy párhuzamos (és még kidolgozandó) informatikai programmal együttesen kerül bevezetésre. Ha ez nem így történik, akkor elkerülhetetlen az olyan torzulás, ami már a most megfogalmazott koncepcióban fellelhető.

A koncepció ugyanis a professzionális személyi számítógépeket úgy említi, mint amiből a magyar ipar a VII. ötéves terv időszakában várhatóan megfelelő kínálatot nyújt. Az import elbírálásában ezért más eszközök beszerzésének kíván prioritást adni. Ismerve a korábbi hazai gyakorlatot, ez magyarul az import nagymértékű hiányát is jelentheti az elkövetkezendőkben. 1985/2. számunkban megjelent helyzetelemzésünkben mi pontosan az ellenkező következtetésre jutottunk.

A következtetés lényege, hogy a kommersz alapberendezések gyártásával a magyar ipar már most zsákutcába jutott. Egyszerűen objektíve lehetetlen, hogy a száz darabos éves termelést akár ezres nagyságrendűre feltornászva, ki tudjon kerülni az 5–10-szeres árszínvonal piacgátló fogságából. Magyarán szólva, a hazánkban az eddigiekben pontosan az import (és azon belül is az utasforgalmú import) miatt megindult mikroszámítógépes lavina a tervezett autarchikus intézkedéssel megreked, amikor a kommersz alapberendezések ösztönzött importjával zökkenésmentesen továbbgördíthető lenne. Arról eddig még nem is szóltunk, hogy a hazai gyártás ezres nagyságrendűre való mesterséges feltornászása néhány gyártónál, ugyanennyi kommersz alapgép devizaköltségénél jóval nagyobb devizaköltséggel lenne csak megvalósítható. Még inkább, mint korábban, az iskolaszámítógép-gyártás esetében.

Mindezek ismeretében nem is az informatika lényegi feltárása tűnik a legsürgetőbb feladatnak, hanem az ezzel összefüggésben levő mikroszámítógépes lavina mozgató erőinek és előrehaladásának feltárása, korrekt bemutatása. Ezt teszszük megkezdett elemző sorozatunkban, ami történelmi módszerrel teljesen érthető képét adja a fejlődésnek. Az egyszerűség kedvéért elsősorban a technológiák és a mikroszámítógépesítés egymásra hatását mutatjuk be, a háttérben azonban mindenütt fellelhető az informatika és a számítástechnika.

NACSA SÁNDOR

LOGO – a programozástanulás új útja

A programépítés változatai

Miután a cikk első részében egy négyzet felrajzolása kapcsán sejtéseink támadtak a programozásról – anélkül, hogy a számítógép elvét előzőleg belénk sulykolták volna –, szemléljük tovább a „marsbéli tájat".

Nézzük meg közelebbről a változó bevezetését. Ahhoz, hogy négyzetünk oldalhosszát könnyen módosítani tudjuk, be kell vezetni az OLDAL változót:

```
TUDD NÉGYZET : OLDAL
  ISMÉTELD 4
  [ELŐRE : OLDAL JOBBRA 90]
KÉSZ
```

A következőkben próbáljunk valami bonyolultabbat rajzolni. Mondjuk, építsünk egy téglafalat. A kőművest irányító algoritmus a következő:

1) Lépj jobbra egy téglányit
2) Végy egy adag habarcsot és terítsd le
3) Végy egy téglát és helyezd a habarcsra
4) Ha kész egy téglasor, menj vissza a fal elejére
5) Ha kész a téglafal, pihenj
6) Egyébként folytasd az 1) lépéssel

Megjegyezzük, hogy egy vérbeli programozó az 5) és 4) lépést végezné el először.

Az alábbiakban két változatban ismertetjük a falépítés és egyszersmind a programépítés folyamatát, LOGO-ban programozva.

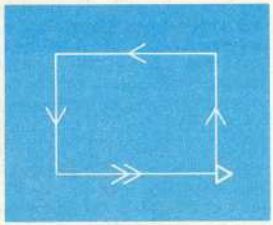
1. változat. Építkezés alulról felfelé

Az alulról felfelé építkezés azt jelenti, hogy először elkészítjük a fal alkotóelemeit, a habarcsot és a téglát, majd a kész elemekből összeállítunk egy téglasort. Ezután már nem lesz nehéz felhúzni a téglafalat.

Kezdjük a habarccsal:

```
TUDD HABARCS : TH
      ELŐRE : TH
KÉSZ
```

A habarcs leterítését egy téglahossznyi (TH) vonal lerajzolásával ábrázoljuk. Most megszerkesztjük a téglát rajzoló eljárást.



1. ábra

```
TUDD TÉGLA : TH : TM
  ISMÉTELD 2
  [BALRA 90 ELŐRE : TM
   BALRA 90 ELŐRE : TH]
KÉSZ
```

Ez az eljárás a NÉGYZET eljáráshoz hasonlít, de annál egy kicsit bonyolultabb, mivel a téglának hossza (TH) és magassága (TM) is van, míg a négyzetnek csak az OLDAL a paramétere.

Most már hívhatjuk a habarcsot és a téglát rajzoló eljárásokat:

```
JOBBRA 90 HABARCS 40
  TÉGLA 40 20
```

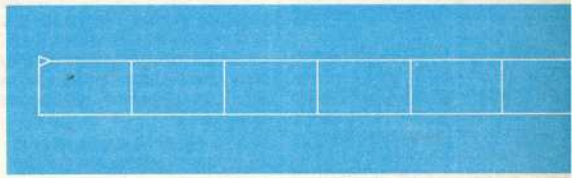
A hívás eredményeként az első utasítás (JOBBRA 90) jobbra fordítja a teknős bekapcsoláskor felfelé mutató orrát, majd a HABARCS eljárás egy 40 lépésnyi vonalat húz. A TÉGLA eljárás ezután lerajzol egy 40 × 20 lépés méretű léglát, melynek alsó éle elnyomja a habarcs vonalát *(1. ábra)*.

Most próbáljunk lerajzolni egy téglasort. Szerkesszük meg az eljárást:

```
TUDD TÉGLASOR : FH : TH
  : TM
  ISMÉTELD : FH
  [HABARCS : TH
  TÉGLA : TH : TM]
  VISSZALÉPÉS : FH : TH : TM
KÉSZ
```

A TÉGLASOR eljárás egymás mellé rajzol annyi téglát, amennyit az FH (falhossz) változóval megadtunk. A VISSZALÉPÉS eljárásra azért van szükség, hogy visszaállítsuk a tenkőst a felépült téglasor legelső téglájának bal felső sarkába, lehetővé téve a következő téglasor építésének elkezdését. Kiegészítőleg szerkesszük meg a VISSZALÉPÉS eljárást:

```
TUDD VISSZALÉPÉS : FH : TH
  : TM
    BALRA 90 ELŐRE : TM
    JOBBRA 90
    ISMÉTELD : FH
    [HÁTRA : TH]
KÉSZ
```



2. ábra

A TÉGLASOR eljárás hívása előtt a teknőst a megfelelő utasítás segítségével a képernyő bal alsó sarkába kell állítani, úgy, hogy a képernyőt jobban ki lehessen használni a téglafal rajzolásához. Egy ilyen előkészület után hívhatjuk a TÉGLASOR eljárást: TÉGLASOR 6 30 15 – mire megjelenik a képernyő alján egy 6 téglából álló téglasor képe, 30 × 15 méretű téglákból felépítve *(2. ábra)*.

Most a teljes téglafalat kellene megrajzolni. Ismét a szerkesztéssel kezdjük:

```
TUDD TÉGLAFAL : FH : FM
  : TH : TM
  ISMÉTELD : FM
  [TÉGLASOR : FH : TH : TM]
KÉSZ
```

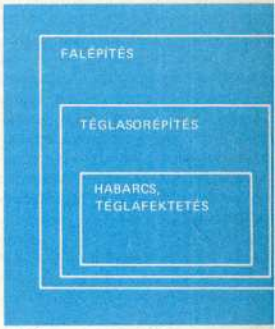
A TÉGLAFAL eljárás tartalmazza az előbbiekben kidolgozott TÉGLASOR eljárást, a falmagasságot (FM) ismét téglaszámmal adjuk meg. Ha most hívjuk az új eljárást (például TÉGLAFAL 6 10 30 15), a LOGO felrajzol a képernyőre egy 6 téglányi hosszú, 10 téglányi magasságú, 30 × 15 lépés méretű téglákból álló téglafalat. Természetesen a hívás előtt a teknőst a képernyő bal alsó sarkába kell állítani, úgy, hogy az orra jobbra mutasson.

2. változat. Strukturált programozás

Most egy képtelen dolgot kísérelünk meg: megpróbálunk felülről lefelé építkezni. Szándékunk érthetőbb lesz, ha azt mondjuk, hogy kiinduláskor a problémát egészében szemléljük, majd lépésről lépésre részleteire bontjuk. A falépítés esetében gondolatmenetünk a következő:

1) Kell építeni egy téglafalat
2) A téglafal téglasorokból áll
3) A téglasor téglákból és habarcsból áll

A *3. ábra* szemlélteti a gondolatmenetet, melyet strukturált gondolatmenetnek is hívhatunk.



3. ábra

Szerkesszük meg a téglafal építését a strukturált gondolatmenetnek megfelelően:

```
TUDD TÉGLAFAL : FH : FM
  : TH : TM
  ISMÉTELD : FM
  [TÉGLASOR : FH : TH : TM]
KÉSZ
```

Most a téglasor építését szerkesztjük meg:

```
TUDD TÉGLASOR : FH : TH
  : TM
  ISMÉTELD : FH
  [HABARCS : TH TÉGLA : TH
  : TM]
  VISSZALÉPÉS
KÉSZ
```

Fentebb már említettük, hogy egy téglasor lerajzolása után a rajzeszközt vissza kell vinni a felépült téglasor elejére, hogy előkészüljünk a következő téglasor megrajzolásához. Erre szolgál az 1. válto-

zatban részletezett VISSZALÉPÉS eljárás, melyre itt csupán hivatkozunk, nem ismételjük meg leírását.

Harmadik lépésben a HABARCS és a TÉGLA eljárásokat kell megszerkeszteni. Ezeket szintén ismertettük az 1. változatban.

Látható, hogy végeredményben ugyanazt a programot építettük fel, csupán megfordítottuk az eljárások beprogramozásának sorrendjét. Ezért felmerül a kérdés: miért van szükség strukturált programozásra?

Ilyen egyszerű, könnyen áttekinthető feladat esetében, mint a falépítés, valóban nélkülözni lehet a strukturált közelítést. Gondoljunk azonban arra, hogy egy bonyolult, sok összetevőből álló, nehezen átlátható belső kapcsolatrendszerrel bíró feladat esetében (például termelésirányítás, mérnöki tervezés) szükség van a strukturált módszer alkalmazására, mert segítségével megbízhatóvá válik a rendszerelemzés és a programozás folyamata. (A strukturált módszer tanulmányozásához jó bevezetést ad Aszalós János: A strukturált programozás irodalmának áttekintése, SZÁMKI Közlemények 19, SZÁMKI, 1978.) Jelen példával csupán a strukturálás elvét kívántuk szemléltetni, a LOGO segítségével.

Következtetések

Célunk az volt, hogy bemutassuk, mi a jelentősége a vizuális visszacsatolásnak a programozás oktatásában, tanulásában. Egyszerű példákon keresztül bizonyítottuk, hogy a szemléltetés lehetőségével felfegyverkezve a programozás gyakorlatilag minden elemét (utasítás, ciklus, eljárás), szerkesztési elveit (építkezés alulról felfelé, strukturált programozás) néhány órás szórakoztató oktatással meg lehet ismertetni az iskolás gyerekekkel és a személyi számítógéppel ismerkedő felnőttekkel. Másik célunk az volt, hogy kiküszöböljük az oktatási folyamatból az elektronikus számítógép elveinek ismertetését, mert úgy gondoljuk, hogy a jövőben a felhasználók többsége számára ez a kérdés érdektelen lesz.

A LOGO nyelv mindkét feladat megoldására kitűnő eszköznek bizonyult. Erősebben is fogalmazhatunk: úgy véljük, hogy a LOGO korszakos jelentőségű az informatikai kultúra tömeges elterjesztését célzó mozgalmak számára.

DR. HROTKÓ GÁBOR

Megoldás rekurzív programozással

Programtervezés során gyakran előfordul, hogy egy-egy részmegoldást legtisztábban és legtömörebben a rekurzió segítségével tudunk kidolgozni. Ez olyan eljárások és függvények bevezetését jelenti, amelyek önmagukat hívják, illetve a függvény meghatározása saját magán alapszik.

Az utóbbira jó példa a faktoriális függvény, amely minden n szám faktoriálisát úgy határozza meg, mint n−1 szám faktoriálisa szorozva n-nel, faktoriális(1) értékét pedig definíció szerint 1-nek tekinti.

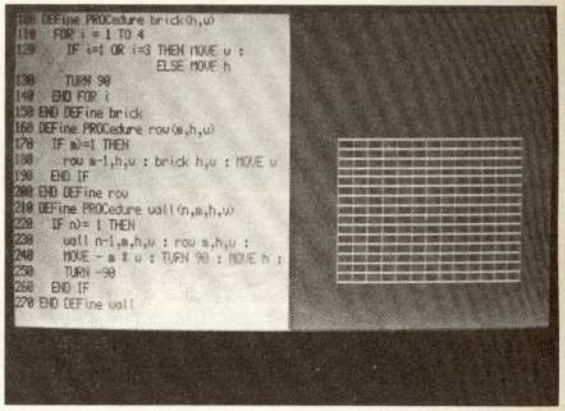
A téglafal-rajzoló feladatra is igen szép rekurzív megoldás dolgozható ki. (Rekurzió segítségével megoldott feladatokkal lapunk 1984/5. számának 14. oldalán és az 1985/2. szám 38. oldalán is találkozhattak az olvasók.)

A téglafal esetében abból kell kiindulnunk, hogy minden téglafal *n* téglasorból áll, minden téglasor *m* téglából, egy tégla pedig *h* magasságú és *w* szélességű. Továbbmenve, minden *n* téglasoros falra igaz, hogy ugyanolyan, mint egy *n−1* téglasoros fal, amire még egy sort helyeztünk. 0 téglasoros fal elvben megengedhető, csak akkor definíció szerint „üres" falat kell „rajzolnunk" (vagyis nem kell rajzolnunk semmit). A téglasorok esetében szintén igaz, hogy egy bármilyen *m* téglányi sor pontosan ugyanolyan, mint egy *m−1* téglányi sor, amit jobbról kiegészítettünk még egy téglával. A 0 téglából álló sor definíciója itt is az „üres" sor (vagyis ismét nem rajzolunk semmit).

A *képünkön* látható általános téglafal-rajzoló programmal (bal oldali ablak) egy 15 sor magas, soronként 13 téglából álló, és 3 egység magasságú, 5 egység szélességű téglákból „felépített" falat (jobb oldali ablak) rajzoltunk ki a wall 15,13,3,5 hívás segítségével (alsó, hosszú ablak). (Az alsó ablak tartalma technikai okokból nem tökéletesen olvasható, ezért itt közöljük:

```
circle 15,13,3,5
cls
pendown
wall 15,13,3,5)
```

A bemutatott megoldás a Sinclair QL típusú számítógép SuperBASIC nyelvén készült. A SuperBASIC az eredeti BASIC nyelv továbbfejlesztett változata. Utasításkészletében megtalálhatók a strukturált programozáshoz szükséges alapvető vezérlési struktúrák, mint a strukturált IF, FOR, REPEAT, a rekurzivan is hívható, illetve paraméterezhető eljárás és függvénydefiníciós utasítások.



```
110 DEFine PROCedure brick(h,w)
120   FOR i = 1 TO 4
130     IF i=1 OR i=3 THEN MOVE w :
                      ELSE MOVE h
140     TURN 90
150   END FOR i
160 END DEFine brick
170 DEFine PROCedure row(n,h,w)
180   IF n>1 THEN
190     row n-1,h,w : brick h,w : MOVE w
200   END IF
210 END DEFine row
220 DEFine PROCedure wall(n,m,h,w)
230   IF n>= 1 THEN
240     wall n-1,m,h,w : row m,h,w :
250     MOVE - m * w : TURN 90 : MOVE h :
260     TURN -90
270   END IF
280 END DEFine wall
```

Az utasítások alakja is megfelel a strukturált programozás követelményeinek, azaz záró END-ek alkalmazása is lehetséges, sőt kötelező. Ezzel és a programsorok képünkön látható, „belsőbb" elhelyezésével a program szövege is teljesen strukturált külalakban jelenik meg. Nagyobb méretű programok készítésénél ezek elengedhetetlen előfeltételei a hatékony fejlesztési munkának. (A téma iránt érdeklődőknek ajánljuk Dahl–Dijkstra–Hoare Strukturált programozás című könyvét, amely a Műszaki Könyvkiadó gondozásában magyarul is megjelent 1978-ban.)

Falrajzoló programunk teljesen „angol nyelvű" változatban készült, azaz a téglát *brick* néven, a sort *row* néven, a falat pedig *wall* néven határoztuk meg. Ezzel is hangsúlyozni kívántuk, hogy a programozás könnyű elsajátításához és későbbi gyakorlásához nagyon fontos az eredeti nyelvi környezethez való kötődés, amit az alapcikk szerzője kiválóan illusztrált. Mi nem tudtunk magyar nyelvű változatot készíteni, mivel más nyelvekhez hasonlóan, a SuperBASIC is angol nyelvű kulcsszavakat használ.

A program értéséhez ismerni kell a SuperBASIC teknőcgrafikai primitíváit is. Ezek általánosabbak és ezért egyszerűbben használhatók, mint a LOGO-ban találhatók. A MOVE utasítás pozitív vagy negatív irányban *mozgatja el* a teknőcgrafika tollát. A TURN pozitív argumentumok esetében az óramutató járásával ellenkező irányban, azaz balra, negatív argumentumok esetében az óramutató járásával megegyező irányban, azaz jobbra *fordítja el* a toll mozgási irányát.

A *row* és *wall* eljárások végén elhelyezett grafikai utasítások a következő tégla, illetve a következő sor megfelelő rajzolását készítik elő. Tudni kell még, hogy a QL bekapcsolása után a toll nem leengedett állapotban, a jobb oldali képernyő bal alsó sarkában van. A CIRCLE 15,15,15 utasítással, annak mellékhatásaként, áthelyeztük a 15,15 koordinátájú pontba, töröltük az ablakot (CLS) és leengedtük a tollat (PENDOWN).

Úgy véljük, hogy a bemutatott program elemzésével a lehető legkönnyebben megfejthető a rekurzió működése a számítógépen. Ezért mindenkinek, aki még nem alkalmazott ilyesmit, ajánljuk, hogy kövesse végig a program működését. Felhívjuk a figyelmet, hogy minden rekurzív hívásnak saját *m*, illetve *n* és *m* paraméterei lesznek.

Ha e program végrehajtása során sikerült teljesen megfejteni a gép SuperBASIC szintű működését, akkor ajánljuk, hogy próbálják ugyanezt a programot „magyar nyelvű", de szintén strukturált változatra átírni. Ezzel egész más irányból közelíthetik meg a strukturált programozás jobb elsajátítását. Egy másik feladat lehet ezek után a faktoriális függvény értékét kiszámító rekurzív függvény meghatározása. Ennek során a RETURN <érték> utasítást lehet használni az éppen kiszámított függvényérték visszaadására SuperBASIC-ben, a DEFine PROCedure helyett pedig a DEFine FUNCtion-t. A függvény aktuálisan kiszámított értéke mindig a hívás helyén adódik vissza, azaz ahova a függvényhívást írtuk, ott <érték> jelenik meg a hívásból való visszatérést követően. Ezzel meg is tehetik az első lépést a rekurzív programozásban. Sok sikert!

(Nino)

COMMODORE

Floppykezelés Kernal-rutinok segítségével

A Commodore 64 terjedelmes irodalma a BASIC nyelvű programozást segíti. A gépi kódú utasításokat többnyire csak felsorolják, pedig szinvonalas feladatmegoldásra csak assembler szintű támogatással van mód.

Cikkünkben az assembler szintű floppykezelést mutatjuk be. A témával kapcsolatban a legtöbb információt az Abacus Software The Anatomy of the Commodore 64 című könyvben találtunk.

Kiemeljük azokat az alapelemeket, amelyekből felépíthető minden – BASIC-ből már ismert, floppykezeléssel kapcsolatos – parancs, illetve utasítás. Szó lesz a SAVE, LOAD, OPEN, I5,8,15,"I",... lemez szintű parancsokról, majd a relatív, szekvenciális és random fájlok eléréséről, de nem térünk ki a gépi kódú utasítások ismertetésére.

A C64 és a floppy közötti kommunikáció a Kernal-rutinokkal történik. Ezek közül a következők fontosak számunkra (eredeti elnevezésükkel):

SETLFS ($FFBA)
A BASIC OPEN utasításának első három paraméterét „közli" a lemezegységgel. Az X regiszterbe a logikai fájl számát (OPEN ③,8,3,...), az Y regiszterbe a (lemez)egység számát (OPEN3, ⑧,3,...), az akkumulátorba a másodlagos címet vagy csatorna számát kell tölteni OPEN3,8,③,...).

SETNAM ($FFBD)
Az elérni kívánt fájl nevét lehet vele kiküldeni. Az akkumulátorba a fájlnév hosszát, az X és Y regiszterekbe a fájlnév címét kell írni. Ha lemez szintű parancsokról van szó, vagyis csak „sima" OPEN15,8,15 történik, az akkumulátorba 0-t kell tölteni, hiszen a fájlnév hossza 0, az X és Y regiszterek tartalma közömbös.

OPEN ($FFC0)
A SETNAM, SETLFS-vel definiált fájlt megnyitja.

CHKOUT ($FFC9)
Ha megtörtént a fájl specifikációja (SETNAM, SETLFS) és megnyitása (OPEN), a megnyitott csatornát kimeneti csatornaként fogja kezelni (a C64-től a floppy felé történő adatforgalmat készíti elő).

CHROUT ($FFD2)
A CHKOUT-tal előkészített csatornán kiküldi az akkumulátorba töltött bájtot.

CHKIN ($FFC6)
A SETNAM, SETLFS, OPEN-nel megnyitott csatornát bemeneti csatornaként definiálja (az adatforgalom a floppytól a C64 felé fog történni – az akkumulátoron keresztül).

CHRIN ($FFCF)
A CHKIN-nel előkészített csatornán behoz egy bájtot az akkumulátorba.

CLRCHN ($FFCC)
A B/K csatornák kezdeti állapotát állítja vissza.

Ki kell adni az azonos típusú floppyműveletek után és/vagy a csatorna bezárása előtt.

CLOSE ($FFC3)
A CLRCHN-nel együtt megegyezik a BASIC CLOSE utasítással. A csatorna számát az akkumulátorba kell tölteni.

LOAD ($FFD5)
A LOAD és a VERIFY műveletek ezzel a Kernal-rutinnal hajthatók végre. Ha a JSR $FFD5 előtt az akkumulátorba 1-et töltünk VERIFY, ha 0-t, LOAD fog történni. Kijelölhető az is, hogy a program az eredeti

```
A9 03       LDA #3     ;LOGIKAI FILE SZAMA
A2 08       LDX #8     ;FLOPPY EGYSEG SZAMA
A0 03       LDY #3     ;MASODLAGOS CIM/CSATORNA
20 BA FF    JSR $FFBA  ;SETLFS RUTIN
AD 66 03    LDA LNEV   ;A FILENEV HOSSZA
A2 67       LDX #<NEV  ;A FILENEV CIMENEK LO BYTE-JA
A0 03       LDY #>NEV  ;A FILENEV CIMENEK HI BYTE-JA
20 BD FF    JSR $FFBD  ;SETNAM RUTIN
```

1. példa

```
A9 01       LDA #1     ;LOGIKAI FILE SZAMA
A2 08       LDX #8     ;FLOPPY EGYSEG SZAMA
A0 01       LDY #1     ;MASODLAGOS CIM(LOAD LESZ AZ EREDETI HELYRE
20 BA FF    JSR $FFBA  ;SETLFS RUTIN
AD 66 03    LDA LNEV   ;A FILENEV HOSSZA
A2 67       LDX #<NEV  ;A FILENEV CIMENEK LO BYTE-JA
A0 03       LDY #>NEV  ;A FILENEV CIMENEK HI BYTE-JA
20 BD FF    JSR $FFBD  ;SETNAM RUTIN
A9 00         LDA #0   ;LOAD LESZ(1-NEL VERIFY LENNE)
20 D5 FF      JSR $FFD5 ;LOAD RUTIN
60            RTS
      LNEV    .BYTE 6
       NEV    .BYTE 80,82,79,66,65,50 ;  0 (ASCII KARAKT.)
A9 01       LDA #1     ;LOGIKAI FILE SZAMA
A2 08       LDX #8     ;FLOPPY EGYSEG SZAMA
A0 01       LDY #0     ;MASODLAGOS CIM(LOAD LESZ EGY MEGADOTT CIMR
20 BA FF    JSR $FFBA  ;SETLFS RUTIN
AD 66 03    LDA LNEV   ;A FILENEV HOSSZA
A2 67       LDX #<NEV  ;A FILENEV CIMENEK LO BYTE-JA
A0 03       LDY #>NEV  ;A FILENEV CIMENEK HI BYTE-JA
20 BD FF    JSR $FFBD  ;SETNAM RUTIN
A9 00         LDA #0   ;LOAD LESZ(1-NEL VERIFY LENNE)
A2 00       LDX #00    ;AZ UJ CIM LO BYTE-JA
A0 C1       LDY #C1    ;AZ UJ CIM HI BYTE-JA
20 D5 FF    JSR $FFD5  ;LOAD RUTIN
96          RTS
      LNEV  .BYTE 6
       NEV  .BYTE 80,82,79,66,65,50 ; 0
```

2. példa

```
A9 01       LDA #1     ;LOGIKAI FILE SZAMA
A2 08       LDX #8     ;FLOPPY EGYSEG SZAMA
A0 01       LDY #1     ;MASODLAGOS CIM
20 BA FF    JSR $FFBA  ;SETLFS RUTIN
AD 66 03    LDA LNEV   ;A FILENEV HOSSZA
A2 67       LDX #<NEV  ;A FILENEV CIMENEK LO BYTE-JA
A0 03       LDY #>NEV  ;A FILENEV CIMENEK HI BYTE-JA
20 BD FF    JSR $FFBD  ;SETNAM RUTIN
A9 C0       LDA #$C0   ;A KEZDOCIM HI BYTE-JA
85 FE       STA $FE    ;0.LAPRA ELHELYEZNI
A9 00       LDA #$00   ;A KEZDOCIM LO BYTE-JA
85 FD       STA $FD    ;
A9 FD       LDA #$FD   ;MUTATO A 0. LAPON
A2 01       LDX #1     ;VEGCIM+1 LO BYTE-JA
A0 C8       LDY #$C8   ;VEGCIM+1 HI BYTE-JA
20 D8 FF    JSR $FFD8  ;SAVE RUTIN
```

3. példa

```
A9 0F       LDA #15    ;LOGIKAI FILE SZAMA
A2 08       LDX #8     ;FLOPPY EGYSEG SZAMA
A0 0F       LDY #15    ;PARANCSCSATORNA
20 BA FF    JSR $FFBA  ;SETLFS RUTIN
A9 00       LDA #$00   ;A FILENEV HOSSZA 0
20 BD FF    JSR $FFBD  ;SETNAM RUTIN
20 C0 FF    JSR $FFC0  ;OPEN RUTIN
```

4. példa

5. példa

```
A9 0F       LDA #15    ;A 15.CSATORNA
20 C9 FF    JSR $FFC9  ;OUTPUT CSATORNA LESZ(CHKOUT RUTIN)
A2 .. ..    LDX HOSSZ  ;AZ UZENET HOSSZA
A0 00       LDY #$00   ;INDEX
B9 .. .. JJ1 LDA UZENET,Y
20 D2 FF    JSR $FFD2  ;CHROUT RUTIN
C8          INY
CA          DEX
D0 F6       BNE JJ1
```

```
A9 0F       LDA #15   ;LOGIKAI FILE SZAMA
A2 08       LDX #8    ;FLOPPY EGYSEG SZAMA
A0 0F       LDY #15   ;PARANCSCSATORNA
20 BA FF    JSR $FFBA ;SETLFSRUTIN
A9 00       LDA #$00  ;A FILENEV HOSSZA 0
20 BD FF    JSR $FFBD ;SETNAM RUTIN
20 C0 FF    JSR $FFC0 ;OPEN RUTIN
A2 0F       LDX #15   ;A 15.CSATORNA OUTPUT CSATORNA LESZ
20 C9 FF    JSR $FFC9 ;CHKOUT RUTIN
A2 09       LDX #9    ;AZ UZENET HOSSZA
A0 00       LDY #$00  ;INDEX
B9 .. .. JJ1 LDA NEV,Y
20 D2 FF    JSR $FFD2 ;CHROUT RUTIN
C8          INY
CA          DEX
D0 F6       BNE JJ1
20 CC FF    JSR $FFCC ;CLRCHN RUTIN
A9 0F       LDA #15   ;A 15.CSATORNA LEZARASA
20 C3 FF    JSR $FFC3 ;CLOSE RUTIN
60          RTS
      LNEV  .BYTE 6
       NEV  .BYTE 83,48,58,80,82,79,66,65,50 ;'S0:PROBA2'
```

6. példa

```
A9 03       LDA #3    ;LOGIKAI FILE SZAMA
A2 08       LDX #8    ;FLOPPY EGYSEG SZAMA
A0 03       LDY #3    ; CSATORNA
20 BA FF    JSR $FFBA ;SETLFSRUTIN
A9 00       LDA LNEV  ;A FILENEV HOSSZA
A2 ..       LDX #<NEV ;A FILENEV CIMENEK LO BYTE-JA
A0 ..       LDY #>NEV ;A FILENEV CIMENEK HI BYTE-JA
20 BD FF    JSR $FFBD ;SETNAM RUTIN
20 C0 FF    JSR $FFC0 ;OPEN RUTIN
  .
  .
  .
  .
  .
      LNEV  .BYTE 10
       NEV  .BYTE 80,82,79,66,65,53,44,83,44,82 ; 'PROBA3,S,R'
```

7. példa

```
A2 03       LDX #3    ;A 3.LOGIKAI FILE
20 C6 FF    JSR $FFC6 ;INPUT CSATORNA LESZ(CHKIN)
20 CF FF    JSR $FFCF ;BEHOZ EGY BYTE-T AZ A-BA(CHRIN)
8D 00 C1    STA $C100
20 CC FF    JSR $FFCC ;KEZDETI ALLAPOTRA(CLRCHN)
```

8. példa

```
A2 03       LDX #3    ;A 3.LOGIKAI FILE
20 C6 FF    JSR $FFC6 ;INPUT CSATORNA LESZ(CHKIN)
A0 00       LDY #0    ;INDEX
20 CF FF JJ1 JSR $FFCF ;BEHOZ EGY BYTE-T AZ A-BA(CHRIN)
C9 0D       CMP #13   ;CR KARAKTER?
F0 07       BEQ JJ2
99 00 C1    STA $C100,Y
C8          INY
4C 00 03    JMP JJ1
20 CC FF JJ2 JSR $FFCC ;KEZDETI ALLAPOTRA(CLRCHN)
```

9. példa

```
A2 03       LDX #3    ;A 3.LOGIKAI FILE
20 C9 FF    JSR $FFC9 ;OUTPUT CSATORNA LESZ(CHKOUT)
A9 41       LDA #65   ;'A'
20 D2 FF    JSR $FFD2 ;CHROUT RUTIN
20 CC FF    JSR $FFCC ;KEZDETI ALLAPOTRA(CLRCHN)
```

10. példa

11. példa

```
      LNEV  .BYTE 1
       NEV  .BYTE 35 ;'#'
A9 03       LDA #3    ;LOGIKAI FILE SZAMA
A2 08       LDX #8    ;FLOPPY EGYSEG SZAMA
A0 03       LDY #3    ;MASODLAGOS CIM/CSATORNA
20 BA FF    JSR $FFBA ;SETLFSRUTIN
AD 66 03    LDA LNEV  ;A FILENEV HOSSZA
A2 67       LDX #<NEV ;A FILENEV CIMENEK LO BYTE-JA
A0 03       LDY #>NEV ;A FILENEV CIMENEK HI BYTE-JA
20 BD FF    JSR $FFBD ;SETNAM RUTIN
20 C0 FF    JSR $FFC0 ;OPEN RUTIN
```

```
        U1 .BYTE 85,49,58,51,32,48,32,49,56,32,48 ;'U1:3 0 18 0'
        BP .BYTE 66,46,80,58,51,32,49,52,52 ;'B-P:3 144'
A2 0F       LDX #15   ;A 15.CSATORNA OUTPUT CSATORNA LESZ
20 C9 FF    JSR $FFC9 ;CHKOUT RUTIN
A2 0B       LDX #11   ;11 KARAKTERT KULDUNK KI
A0 00       LDY #0    ;INDEX
B9 83 03 JJ1 LDA U1,Y
20 D2 FF    JSR $FFD2 ;CHROUT RUTIN
C8          INY
CA          DEX
D0 F6       BNE JJ1
20 CC FF    JSR $FFCC ;CLRCHN RUTIN
A2 0F       LDX #15   ;A 15.CSATORNA OUTPUT CSATORNA LESZ
20 C9 FF    JSR $FFC9 ;CHKOUT RUTIN
A2 09       LDX #9
A0 00       LDY #0
B9 94 03 JJ2 LDA BP,Y
20 D2 FF    JSR $FFD2 ;CHROUT RUTIN
C8          INY
CA          DEX
D0 F6       BNE JJ2
20 CC FF    JSR $FFCC ;CLRCHN RUTIN
```

12. példa

```
A9 03       LDA #3    ;LOGIKAI FILE SZAMA
A2 00       LDX #0    ;FLOPPY EGYSEG SZAMA
A0 03       LDY #3    ; CSATORNA
20 BA FF    JSR $FFBA ;SETLFSRUTIN
A9 00       LDA LNEV  ;A FILENEV HOSSZA
A2 ..       LDX #<NEV ;A FILENEV CIMENEK LO BYTE-JA
A0 ..       LDY #>NEV ;A FILENEV CIMENEK HI BYTE-JA
20 BD FF    JSR $FFBD ;SETNAM RUTIN
20 C0 FF    JSR $FFC0 ;OPEN RUTIN
  .
  .
  .
  .
  .
      LNEV  .BYTE 9
       NEV  .BYTE 82,69,76,65,44,76,44,28 ;'RELA,L,+CHR$(28)'
```

13. példa

```
A2 0F       LDX #15   ;A 15.CSATORNA OUTPUT CSATORNA LESZ
20 C9 FF    JSR $FFC9 ;CHKOUT RUTIN
A0 00       LDY #0
A2 05       LDX #5
B9 A9 03 JJ1 LDA PERT,Y
20 D2 FF    JSR $FFD2 ;CHROUT RUTIN
C8          INY
CA          DEX
D0 F6       BNE JJ1
20 CC FF    JSR $FFCC   ;CLRCHN RUTIN
  .
  .
  .
  .
      PERT  .BYTE 00,3,2,8,1 ; 'PCHR$(3)CHR$(2)CHR$(0)CHR$(1)'
```

14. példa

```
10 FOR I=0TO48:READA:POKE849+I,A:NEXT:GOTO50
20 DATA 169,1,162,8,160,1,32,186,255,173,64,3,162,65,160,3,32,189,255
30 DATA 173,61,3,133,254,173,60,3,133,253,169,253,174,62,3,
   172,63,3,32,216,255
40 DATA 96
50 INPUT"KEZDOCIM(DECIMALISAN):";A1
60 I1=INT(A1/256):I2=A1-I1*256
70 POKE826,I2:POKE828,I1
80 INPUT"VEGCIM(DECIMALISAN):";A2
90 I3=INT(A2/256):I4=A2-I3*256
100 POKE830,I4:POKE831,I3
110 INPUT"A FILE NEVE!";A$
120 A=LEN(A$):POKE832,A
130 FOR I=1TOA:POKE832+I,ASC(MID$(A$,I,1)):NEXT
140 SYS849
150 END
```

15. példa

helyére (ahonnan kimentették), vagy egy paraméterként megadott címre töltődjön-e be. Az első esetben a SETLFS-ben a másodlagos címnek kell lennie. A második esetben a másodlagos cím 0, és azt a címet, ahova be kell tölteni a programot, X és Y regiszteren keresztül kell közölni.

SAVE ($FFD8)

A fájlspecifikáció (SETLFS, SETNAM) megtörténte után az akkumulátorba egy 0. lapon lévő cím mutatóját kell írni, ez tartalmazza az első elmentendő bájt címét. Az X és Y regiszterekbe az első, már nem elmentendő bájt címe kerüljön.

Bármely floppyval való művelet az *1. példával* kezdődik.

```
LEMEZNEV.....PAGE 0001

LINE# LOC   CODE        LINE

00001  0000                        #=$033C
00002  033C  A9 0F                  LDA #15
00003  033E  A2 08                  LDX #8
00004  0340  A0 0F                  LDY #15
00005  0342  20 BA FF               JSR $FFBA
00006  0345  A9 00                  LDA #0
00007  0347  20 BD FF               JSR $FFBD
00008  034A  20 C0 FF               JSR $FFC0
00009  034D  A9 03                  LDA #3
00010  034F  A2 08                  LDX #8
00011  0351  A0 03                  LDY #3
00012  0353  20 BA FF               JSR $FFBA
00013  0356  AD C2 03               LDA LNEV
00014  0359  A2 D7                  LDX #<NEV
00015  035B  A0 03                  LDY #>NEV
00016  035D  20 BD FF               JSR $FFBD
00017  0360  20 C0 FF               JSR $FFC0
00018  0363  A2 0F                  LDX #15
00019  0365  20 C9 FF               JSR $FFC9
00020  0368  A0 00                  LDY #0
00021  036A  A2 0B                  LDX #11
00022  036C  B9 C3 03   DUDU        LDA U1,Y
00023  036F  20 D2 FF               JSR $FFD2
00024  0372  C8                     INY
00025  0373  CA                     DEX
00026  0374  D0 F6                  BNE DUDU
00027  0376  20 CC FF               JSR $FFCC
00028  0379  A2 0F                  LDX #15
00029  037B  20 C9 FF               JSR $FFC9
00030  037E  A0 00                  LDY #0
00031  0380  A2 09                  LDX #9
00032  0382  B9 CE 03   JJJ1        LDA BP,Y
00033  0385  20 D2 FF               JSR $FFD2
00034  0388  C8                     INY
00035  0389  CA                     DEX
00036  038A  D0 F6                  BNE JJJ1
00037  038C  20 CC FF               JSR $FFCC
00038  038F  A2 03                  LDX #3
00039  0391  20 C6 FF               JSR $FFC6
00040  0394  A2 10                  LDX #16
00041  0396  A0 00                  LDY #0
00042  0398  20 CF FF   JJJ3        JSR $FFCF
00043  039B  99 00 C1               STA $C100,Y
00044  039E  C8                     INY
00045  039F  CA                     DEX
00046  03A0  D0 F6                  BNE JJJ3
00047  03A2  20 CC FF               JSR $FFCC
00048  03A5  A9 0F                  LDA #15
00049  03A7  20 C3 FF               JSR $FFC3
00050  03AA  A9 03                  LDA #3
00051  03AC  20 C3 FF               JSR $FFC3
00052  03AF  A2 00                  LDX #0
00053  03B1  A0 10                  LDY #16
00054  03B3  BD 00 C1   JJJ2        LDA $C100,X
00055  03B6  C9 A0                  CMP #160
00056  03B8  F0 07                  BEQ VEGE
00057  03BA  20 D2 FF               JSR $FFD2
00058  03BD  E8                     INX
00059  03BE  88                     DEY
00060  03BF  D0 F2                  BNE JJJ2
00061  03C1  60            VEGE    RTS
00062  03C2  01            LNEV    .BYTE 1
00063  03C3  55            U1      .BYTE 85,49,58,51,32,46,32,49,56,32,48
00063  03C4  31
00063  03C5  3A
00063  03C6  33
00063  03C7  20
00063  03C8  30
00063  03C9  20
00063  03CA  31
00063  03CB  38
00063  03CC  20
00063  03CD  30
00064  03CE  42            BP      .BYTE 66,46,80,58,51,32,49,52,52
00064  03CF  2E
00064  03D0  50
00064  03D1  3A
00064  03D2  33
00064  03D3  20
00064  03D4  31
00064  03D5  34
00064  03D6  34
00065  03D7  23            NEV     .BYTE 35
00066  03D8                        .END
```

16. példa

A programok betöltését, ellenőrzését, elmentését a *2. példában* látható kiegészítésekkel lehet elvégezni.

A program mentése a $C000-tól $C800-ig (SAVE) a *3. példa* szerint végezhető.

A lemez szintű parancsok, utasítások kiküldése a 15. csatornán keresztül történik. A 15. csatorna megnyitása abban különbözik a többi csatorna megnyitásától, hogy itt nincs fájlnév *(4. példa)*.

A parancsok ezután a PRINT#15,"..."-vel adhatók ki. Ennek megfelelője az *5. példa*.

Az UZENET lehet például "I" (inicializálás), "V" (validálás), "SO:..." (fájl törlése), de alakra teljesen megegyezik a fájlok kezelésénél használt "U1:"..., "B-A", "B-P:",... parancsok kiküldésével is. A "PROBA2" fájl törlése így a *6. példa* szerint hajtható végre.

A háromféle adatfájl elérése nagyon hasonló a LOAD, illetve SAVE parancsoknál leírtakhoz. A különbség a fájlnév formájában van – a fájlnévnek tartalmaznia kell a fájl típusára vonatkozó információt is, szekvenciális fájlnál a végrehajtandó műveletre való utalást (olvasásra, írásra nyitottuk-e meg). A BASIC utasításokban is így történik, hiszen az OPEN után idézőjelbe téve le kell írni mindazt, ami a floppy számára egyértelművé teszi, hogy szekvenciális, random vagy relatív fájlról van-e szó, és szekvenciális fájlnál ki kell egészíteni az R,W,A betűk valamelyikével

(OPEN3,8,2"SZEKV,S,R" – OPEN3,8,3"REL,L." CHR (rekordhossz) – OPEN3,8,3,"#")

A szekvenciális fájl a *7. példa* alapján nyitható meg olvasásra. Ez megegyezik a BASIC OPEN3,8,3,"PROBA3,S,R" utasításával. A GET#3,A megfelelője a *8. példa* (egy bájt olvasása a fájlból).

Az INPUT#3,A utasítást meg kell szervezni, a GET#3-t ciklusba kell építeni, és figyelni kell a CR-t (kocsi vissza, soremelés) *(9. példa)*.

Ha a szekvenciális fájl írásra lett megnyitva, a PRINT#3," " vagyis egy bájt kiküldése a floppyra, a *10. példán* látható.

Random fájl megnyitásánál a fájlnév egy "#" jel. Az OPEN3,8,3,"#" gépi kódú megfelelője a *11. példa*.

Az elvégzendő műveletet a parancs-csatornán kell közölni. A 15. csatornát is meg kell nyitni a random fájl használatánál.

A műveletek, amelyek ennél a fájltípusnál alkalmazhatók, a "B-A:", "B-F:", "B-R:", "B-W:", "U1:", "U2:", "B-P:". Ezek közül kettőt „fordítunk" le gépi kódra a *12. példában*. Ugyanaz a forma, mint a lemez szintű parancsoknál.

A pufferben lévő adatokat ugyanazzal a GET rutinnal lehet beolvasni, mint amit a szekvenciális fájl használatánál láttunk. A pufferbe írni az ott mutatott PRINT rutinnal lehet.

Relatív fájl megnyitását: (OPEN3,8,3,"RELA,L" CHR$ (20)) mutatja a *13. példa*.

Az egyes rekordokhoz való hozzáférés a PRINT#15 CHR$ (csatornaszám) CHR$ (rekordszám L0 bájt) CHR$ (rekordszám HI bájt) CHR$ (rekordon belüli pozíció) utasítással történik. Ennek assembler megfelelője hasonló az "U1:", "B-P:", "I", ... utasítások gépi kódú megoldásához, a különbség az, hogy a CHR-n belüli értékeket közvetlenül és egymás után (nem kell köztük szóköz, mivel sem vessző, sem pontosvessző nem választja el a kiküldendő karaktereket) kell kiküldeni a floppynak. Az összes felsorolt utasítás közül ez az egyetlen, amelynél a csatornaszámot és a többi információt nem mint az egyes számjegyek ASC értékét, hanem magát a számot várja a floppy *(14. példa)*.

Végül két kis programot mutatunk be, amelyekben a fent leírtakat alkalmaztuk. Az egyikkel tetszőleges memóriaterület tartalma menthető ki programfájlba, hosszadalmas PEEK utasítássorozatot lehet vele gyorsítani *(15. példa)*. A második a lemez nevét olvassa el és írja ki a képernyőre *(16. példa)*.

A Kernal-rutinok teljes magyar nyelvű leírása megtalálható dr. Úry László Commodore 64 című könyvében (Bp. 1984. LSI ATSZ).

FREY JUDIT

PRIMO

Szótagolás

```
10 REM Szótagolás
20 PRINT "A szótagokat kötőjellel elválasztva"
30 PRINT "kell beírni."
40 PRINT : PRINT : PRINT "Például mosolyog = mo-so-lyog"
50 PRINT : PRINT "Kisbetűkkel kell írni!" : PRINT
60 READ A$
70 IF A$="VÉGE" END
80 READ B$
90 PRINT "Hogy írják szótagolva azt, hogy ";A$
100 INPUT Z$
110 IF Z$=B$ THEN 150
120 PRINT "HIBÁS!"
130 PRINT "Helyesen ";B$
140 GOTO 60
150 PRINT "HELYES!"
160 GOTO 60
200 DATA fiai,fi-a-i,krumpli,krump-li,messze,mesz-sze,vonzza,vonz-za,
gallyat,galy-lyat,rontja,ront-ja,dolgozik,dol-go-zik,VÉGE
```

A program segítségével a szótagolást lehet gyakorolni. A gyakorlásra szánt szavakat a minta szerint párosával, „DATA" utasításban kell megadni. Az utolsó „DATA" utasításban közölt utolsó érték a „VÉGE" szó legyen. A program Primóra készült, a gyakorlásra szánt szavak alkalmas megválasztásával sikeresen használható a magyar ábécét nem ismerő gépeken is.

SOMOGYI GYÖRGY

Display program

A program fő érdekessége a 28 soros képernyő, amit csak gépi kódban lehet kezelni. Négy sornál többet azért nem tanácsos hozzátoldani a 24-hez, mert ekkor a kép futni kezd.

A program másik érdekessége, hogy a kép másodpercenként invertálódik. Ez basicben is működik egészen addig, amíg egy új basic sort be nem írunk, vagy ki nem törlünk. De kész basic program futása közben működik, tehát tulajdonképpen két program fut egyszerre. Ez úgy lehetséges, hogy képernyőkezeléskor a processzor a 0066 ROM címre ugrik, ahol néhány lépés után JP (IX) utasítás található. Ha az IX regisztert átírjuk, a képernyőkezelés után saját programot írhatunk. Így a két programunk egymástól függetlenül egyszerre futhat. Mivel a képernyőkezelés másodpercenként pontosan 50-szer történik, a program felhasználható időzítésre, óra készítésére is.

A program betöltése legkönnyebben Assembler fordítóprogram segítségével lehetséges. A program elején a return a fordítóprogramból való visszatéréshez szükséges. A program később ezt, valamint az utána következő bájtot saját változóként használja.

```
4084  C9 00 2A 0C 40 11 00 50
408C  01 19 03 ED B0 EB 0E 04
4094  06 20 36 00 23 10 FB 36
409C  76 23 0D 20 F3 36 40 2A
40A4  0C 40 E5 21 00 50 22 0C
40AC  40 DD 21 24 41 21 2A 51
40B4  E5 36 88 3E 10 32 7B 40
40BC  1E 30 CD BB 02 44 4D 51
40C4  14 28 16 1D 20 F4 3A 7B
40CC  40 3D 32 7B 40 20 E9 CD
40D4  F0 40 3E 02 32 7B 40 18
40DC  DF CD BB 02 44 4D 51 14
40E4  28 F7 CD F0 40 3E 10 32
40EC  7B 40 18 CC CD BD 07 7E
40F4  FE 0C 28 20 FE 77 CA A5
40FC  41 FE 76 CA B5 41 CB B7
4104  C1 E1 77 23 3E 76 BE 20
410C  06 23 3E 40 BE 28 07 36
4114  88 E5 C5 C9 E1 E1 E1 22
411C  0C 40 DD 21 3B 41 CF 00
4124  ED 5F 01 01 1D 3E F5 CD
412C  B5 02 2B CD 92 02 CD 20
4134  02 DD 21 24 41 18 15 ED
413C  5F 01 01 19 3E F5 CD B5
4144  02 2B CD 92 02 CD 20 02
414C  DD 21 3B 41 F5 C5 D5 E5
4154  3A 84 40 3D 32 84 40 20
415C  0D 3E 32 32 84 40 3A 85
4164  40 EE 01 32 85 40 21 84
416C  40 7E 2F E6 07 20 2B 23
4174  CB 46 28 14 CD CB 41 2A
417C  0C 40 0E 20 23 CB FE 0D
4184  20 FA 23 05 20 F4 18 12
418C  CD CB 41 2A 0C 40 0E 20
4194  23 CB BE 0D 20 FA 23 05
419C  20 F4 E1 D1 C1 F1 C3 A4
41A4  02 C1 E1 36 00 2B 3E 76
41AC  BE 20 01 2B 36 88 E5 C5
41B4  C9 C1 E1 36 00 3E 76 23
41BC  BE 20 FC 23 3E 40 BE CA
41C4  1A 41 36 88 E5 C5 C9 06
41CC  18 2A 0C 40 11 00 50 A7
41D4  ED 52 C0 06 1C C9 DD 21
41DC  3B 41 C9 C3 3C 7A 76 00
```

Gépi kód

Betöltő

```
  1 REM 350 DB SPACE
 100 POKE 16514,118
 110 POKE 16515,118
 120 FOR X=16516 TO 16862
 130 SCROLL
 140 PRINT X;
 150 INPUT A$
 160 PRINT " ";"+A$
 170 POKE X,(CODE A$(1)-28)*16+C
ODE A$(2)-28
 180 NEXT X
```

```
        RET                     KL      LD A,R
        NOP                             LD BC,1901
        LD HL,(400C)                    LD A,F5
        LD DE,5000                      CALL 0285
        LD BC,0319                      DEC HL
        LDIR                            CALL 0292
        EX DE,HL                        CALL 0220
        LD C,04                         LD IX,KL
KX      LD B,20                 IN      PUSH AF
KY      LD (HL),0                       PUSH BC
        INC HL                          PUSH DE
        DJNZ KY                         PUSH HL
        LD (HL),76                      LD A,(4084)
        INC HL                          DEC A
        DEC C                           LD (4084),A
        JR NZ KX                        JR NZ LX
        LD (HL),40                      LD A,32
        LD HL,(400C)                    LD (4084),A
        PUSH HL                         LD A,(4085)
        LD HL,5000                      XOR 01
        LD (400C),HL                    LD (4085),A
        LD IX,KK                LX      LD HL,4084
        LD HL,512A                      LD A,(HL)
        PUSH HL                         CPL
        LD (HL),88                      AND 07
        LD A,10                         JR NZ YY
        LD (407B),A                     INC HL
XX      LD E,30                         BIT 0,(HL)
XY      CALL 02BB                       JR Z MM
        LD B,H                          CALL MX
        LD C,L                          LD HL,(400C)
        LD D,C                  LB      LD C,20
        INC D                   LA      INC HL
        JR Z XA                         SET 7,(HL)
        DEC E                           DEC C
        JR NZ XY                        JR NZ LA
        LD A,(407B)                     INC HL
        DEC A                           DEC B
        LD (407B),A                     JR NZ LB
        JR NZ XX                        JR YY
        CALL PR                 MM      CALL MX
        LD A,02                         LD HL,(400C)
        LD (407B),A             LD      LD C,20
        JR XX                   LC      INC HL
XA      CALL 02BB                       RES 7,(HL)
        LD B,H                          DEC C
        LD C,L                          JR NZ LC
        LD D,C                          INC HL
        INC D                           DEC B
        JR Z XA                         JR NZ LD
        CALL PR                 YY      POP HL
        LD A,10                         POP DE
        LD (407B),A                     POP BC
        JR XX                           POP AF
PR      CALL 07BD                       JP 02A4
        LD A,(HL)               RU      POP BC
        CP 0C                           POP HL
        JR Z BS                         LD (HL),0
        CP 77                           DEC HL
        JP Z RU                         LD A,76
        CP 76                           CP (HL)
        JP Z NL                         JR NZ RX
        RES 6,A                         DEC HL
        POP BC                  RX      LD (HL),88
        POP HL                          PUSH HL
        LD (HL),A                       PUSH BC
        INC HL                          RET
        LD A,76                 NL      POP BC
        CP (HL)                         POP HL
        JR NZ KA                        LD (HL),0
        INC HL                          LD A,76
        LD A,40                 NA      INC HL
        CP (HL)                         CP (HL)
        JR Z BZ                         JR NZ NA
KA      LD (HL),88                      INC HL
        PUSH HL                         LD A,40
        PUSH BC                         CP (HL)
        RET                             JP Z BZ
BS      POP HL                          LD (HL),88
        POP HL                          PUSH HL
BZ      POP HL                          PUSH BC
        LD (400C),HL                    RET
        LD IX,KL                MX      LD B,18
        RST 08                          LD HL,(400C)
        00                              LD DE,5000
KK      LD A,R                          AND A
        LD BC,1D01                      SBC HL,DE
        LD A,F5                         RET NZ
        CALL 0285                       LD B,1C
        DEC HL                          RET
        CALL 0292               TT      LD IX,KL
        CALL 0220                       RET
        LD IX,KK
        JR IN
```

```
    COPY
  1 REM
  2 REM
  3 REM RAND USR 16858
 10 PRINT AT 0,7;"DISPLAY PROGR
AM"
 20 PRINT AT 2,0;"PAPP ZOLTAN B
UDAPEST 1984.VII."
 30 PRINT "SZABADON GEPELHET A
KEPERNYORE  AMELY 28 AZAZ HUSZON
NYOLC SORBOLALL. A KEPERNYO KOZE
EN MASODPER-CENKENT INVERTALODIK
.EZ BASICBE VALO VISSZATERESKOR
SEM VALTOZIK"
 40 SAVE "DISPLA"
 50 RAND USR 16518
```

Assembler fordító

A fordítóprogram az 1 REM-ben tárolja a gépi kódot, 2 REM-ben pedig a CHR$ alakot.

Ha nem áll rendelkezésre Assembler fordító, akkor a programot hexadecimális formája alapján a betöltőprogrammal lehet beírni.

PAPP ZOLTÁN

Lemeznév-változtatás

Lemezrendezés közben sokszor jó lenne, ha a lemez nevét úgy tudnánk megváltoztatni, hogy a programok és az adatok megmaradjanak a lemezen. Ezt a rendelkezésre álló szoftverrel nem tudjuk megoldani. A PRINT#15, "N0:név" paranccsal ugyan tudunk új nevet adni a lemeznek, de ez a parancs a DIRECTORY-t (tartalomjegyzéket) is törli, azaz a lemezen levő adatok a parancs végrehajtása után nem érhetők el. A probléma megoldására a nyugatnémet Commodore újság 1984. 5. számában találtuk az alábbi kis programot, amely hasznos kiegészítése lehet minden programkönyvtárnak.

A program indítása után megjelenik a kérdés, hogy bent van-e a lemez a lemezmeghajtóban. Nemleges válasz esetén ("N") a gép megszakítja a programot és kiírja a CONT szót (a RETURN gomb lenyomásával bármikor folytatható). Ha a feltett kérdésre „I"-vel válaszolunk, akkor a képernyőn megjelenik a meghajtóban levő lemez neve és ID-je, majd pedig a kérdés, hogy lesz-e lemeznévváltozás. Igenlő válasz esetén be kell adni az új nevet. A következő kérdés az, hogy ID-változás lesz-e. Ha igen, meghatározandó az új ID. A lemeznév 16 karakter, az ID pedig 2 karakter hosszú lehet. Ennél hosszabb nevekből a felesleges karaktereket levágja a program. Ha nem használjuk ki a 16, illetve 2 karaktert, akkor

```
 1 REM ********************************
 2 REM *
 3 REM *   DISKETTNEV-VALTOZAS
 4 REM *
 5 REM ********************************
 6 :
1000 PRINT"■■   =-=   DISKETTNEVVALTOZAS  =-=          "
1020 INPUT"■■ENT VAN A DISKETT A TARBAN   I■■■";W$
1030 IF W$<>"I" THEN PRINT"■CONT■■■":END
1040 PRINT"■DISK NEV:";SPC(19);"DISK ID:";PRINTTAB(18);"■■"
1050 OPEN1,8,15:PRINT#1,"I0"
1060 OPEN8,8,8,"#"
1070 PRINT#1,"U1:";8;0;18;0
1080 PRINT#1,"B-P:";8;144
1090 FOR I=1 TO 16
1100 GET#8,CH$:PRINT CH$;
1110 NEXT I
1120 PRINT#1,"B-P:";8;162
1130 GET#8,C1$,C2$
1140 PRINT SPC(11);C1$;C2$
1150 INPUT"■■LESZ NEVVALTOZAS   I■■■";W$:GX=0
1160 IF W$<>"I" THEN1210
1170 INPUT"■UJ NEV ";DN$
1180 DN$=LEFT$(DN$+"                ",16)
1190 PRINT#1,"B-P:";8;144
1200 PRINT#8,DN$:GX=1
1210 INPUT"■■LESZ ID VALTOZAS   N■■■";W$:RX=0
1220 IF W$<>"I" THEN1270
1230 INPUT"■UJ ID ";ID$
1240 ID$=LEFT$(ID$+"  ",2)
1250 PRINT#1,"B-P";8;162
1260 PRINT#8,ID$:RX=1
1270 IFGX=1ORRX=1THEN PRINT#1,"U2:";8;0;18;0
1280 PRINT#1,"I0":CLOSE8:CLOSE1
1290 PRINT"■■■";:IFGX=1THEN PRINT " DISKETTNEV MEGVALTOZOTT -=-";
1300 PRINT" VEGE"
1310 END
```

szóközökkel egészíti ki a program a neveket. Ha a lemez neve megváltozott, akkor a program ezt jelzi a képernyőn.

A program a lemez 18. sávján a 0. szektoron található két információt cseréli ki. A lemez neve a 144–161 bájtig található, az ID pedig a 162, 163 bájton van.

A program a következőképpen működik:

1050 parancs-csatorna nyitása
1060 puffercsatorna nyitása
1070 a tartalomjegyzék beolvasása a pufferbe
1080 a mutatónak a lemeznév első bájtjára állítása
1090–1110 a lemeznév beolvasása és kiírása a képernyőre
1120 a mutatónak az ID első bájtjára állítása
1130–1140 ID beolvasása és kiírása, a mutató a lemeznév első bájtjára áll
1190 a mutató a lemeznév első bájtjára áll
1200 új lemeznév beírása a pufferbe
1250–1260 a mutató az ID-re áll és új ID kiíratása a pufferbe
1270 megváltozott tartalomjegyzék kiírása a pufferből a lemezre
1280 a puffer- és parancs-csatorna zárása

RICHTER JÖRG

Magyar szöveg nyomtatása

Az alábbi BASIC program grafikus mátrixnyomtatón magyar (ékezetes betűkódokat is tartalmazó) szövegek kiírására alkalmas. Állítható a sortávolság is: a két lehetséges érték 1/6 inch (egy normál sor) vagy 5/18 inch (egy grafikus + egy normál sor), ezenkívül vezérli a lapdobásokat is.

A program Commodore 64 gépre és GP–100 VC nyomtatóra készült. Lapemelés-jelzésként egy nulla bájtsorozatot vár a bemeneti fájlban. (Ilyen lapemelés-jelzést ad az Easy Script nevű szövegszerkesztő.) Látható, hogy a program egyáltalán nem „hordozható", de nem nehéz átírni más esetre.

A három feladat (ékezetes betűk, sorváltás, lapdobás) jól elkülöníthető.

Az ékezetes betűk

A mátrixnyomtató minden karaktert 7 × 5 pontból rak össze. „Normál" módban a számítógép felől ASCII-kódokat vár, a karakterek alakját saját ROM-jából olvassa ki. Grafikus módban minden pontra meg kell adni, hogy fekete legyen-e, vagy fehér. Egy ékezetes betű megadásához öt bájt kell. (Csak az alsó hét értékes, a nyolcadik mindig egy legyen.)

```
100 rem 1984 december 14.
110 rem ekezetes Printelo Program
120 :
130 nk=6 : rem ekezetes, sPecialis karakterek szama
140 dim ek$(nk),as%(255)
150 :
160 rem ekezetes betuk adatai
170 data 56,69,68,69,56,0     : rem oe
180 data 58,64,64,58,64,0     : rem ue
190 data 32,84,86,81,64,0     : rem aa
200 data 56,84,86,85, 8,0     : rem ee
210 data 56,68,71,68,56,0     : rem oo
220 data  0, 7, 0, 7, 0,0     : rem "
230 restore : for i=1 to nk : ek$(i)="":
240 for j=1 to 6 : read x : ek$(i)=ek$(i)+chr$(x+128) : next j
250 next i
260 rem karakter - funkcio vektor
270 for i=0 to 255 : as%(i)=0 : next i
280 as%(asc("I"))=1 : as%(asc("J"))=2
290 as%(asc("É"))=3 : as%(asc("$"))=4
300 as%(asc("ß"))=5 : as%(34)=6
310 as%(15)=-1 : as%(17)=-1
320 as%(13)=-2
330 :
340 gosub 380 : rem init
350 goto 670 : rem fociklusba
360 :
370 :
380 rem init
390 Poke 53280,6 : Poke 53281,6 : Print chr$(5) :
    rem szinek beallitasa
400 Printchr$(147)+chr$(14) : rem kePernyotorles, kisbetuk
410 Print"Easy scriPt file Printelese":Print
420 Print:inPut"Forras e9yse9  8■■■";e2
430 if e2<8 or e2>11 goto 420
440 inPut"Forras file ";f2$
450 oPen 15,e2,15 : oPen 2,e2,2,f2$+",s,r"
460 gosub 590 : rem discerr
470 :
480 oPen4,4,7 : Print#4,chr$(15) : rem normal mod
490 Print:Print"Allitsd a Printert laP tetejere !":Print
500 :
510 Print:Print"normal sorvaltas = 1"
520 Print"9rafikus+normal  = 2" : inPut"  2■■■";mo
530 if mo<>1 and mo<>2 goto 510
540 :
550 ln=0 : rem sorszamlalo
560 return
570 :
580 :
590 rem discerr
600 inPut#15,h1,h2$,h3,h4
610 if h1=0 then return
620 Print:Printchr$(18)+chr$(150)+"Disk error: "+chr$(146);h1,h2$
630 Print"sav =";h3,"szektor =",h4,chr$(5)
640 stoP
650 :
660 :
670 rem ez a fociklus
680 :
690 9et#2,a$ : if st<>0 goto 1080 : rem ve9e a filenak ?
700 :
710 rem ve9e a laPnak ?
720 if a$<>"" goto 780 : rem u9rik ha nem
730 9et#2,a$ : ss=st
740 if a$="" and ss=0 then 730 : rem van me9 nulla byte?
750 gosub 920 : rem laPvaltas
760 if ss<>0 goto 1120 : rem file ve9e?
770 :
```

```
780 b%=as%(asc(a$)) : if b%=0 then Print#4,a$; : 9oto 690 :
   rem normal karakter
810 :
820 if b%>0 then Print#4,chr$(8);ek$(b%);chr$(15); : 9oto 690 :
   rem ekezetes
822 :
824 if b%=-1 then 690 : rem no oPeration
830 :
840 if b%<>-2 then Print:Print"Helytelen as% k&d:";b% : 9oto 1120
850 :
860 rem -2, kocsivissza
870 if mo=2 then Print#4,chr$(8)
880 Print#4,chr$(15)
890 ln=ln+1 : 9oto 690
900 :
910 :
920 rem laPvaltas
930 Print:Print"sorok szama =";ln
940 if mo=2 9oto 1000
950 if ln=72 9oto 980
960 if ln>72 then Print:Print"Tobb mint 72 !":Print : 9oto 980
970 for i=1+1 to 72 : Print#4 : next i
980 ln=0 : return
990 :
1000 if ln>42 then Print:Print"Tobb mint 42 !":Print : 9oto 1060
1010 z=216-5*ln : if z<>2*int(z/2) 9oto 1030
1020 9s=z/2 : ns=0 : 9oto 1040
1030 9s=(z-3)/2 : ns=1
1040 Print#4,chr$(8); : for i=1 to 9s : Print#4 : next i
1050 Print#4,chr$(15); : if ns then Print#4
1060 ln=0 : return
1070 :
1080 :
1090 rem file ve9e
1100 9osub 920 : rem laPvaltas
1110 :
1120 rem
1130 close 2 : close 4 : close 15
1140 Print Printtab(13);"end !"
1150 stoP
```

A programban minden ékezetes betűre ezt az öt bájtot az ek string tömb egy eleme tartalmazza. Az idézőjel karaktert is grafikus módban írjuk ki. Ennek oka az, hogy a C64 géphez készült nyomtatók normál módban egy idézőjel után, a következő idézőjelig a vezérlőkaraktereket is kinyomtatják (így a normál – grafikus átkapcsoló karaktert is), valamilyen inverz grafikus karakterként.

Az ékezetes betűket valamilyen egyébként ritkán használt karakterekhez lehet hozzárendelni, ezek helyett írja ki őket a program. Az AS% tömb nulla elemei normál nyomtatást jelentenek, a −1 érték azt jelenti, hogy a karaktert ki kell hagyni, a −2 a soremelő karakter jele. A pozitív, nem nulla értékek azt jelzik, hogy az illető speciális karakter definíciója hányadik DATA-ban volt megadva. Könnyen lehet a programot további ékezetes és speciális karakterekkel bővíteni, ez a futási időt nem növeli.

A soremelés

A szöveg tagoltabb, olvashatóbb, ha nincs túl sűrűn nyomtatva, a normál soremelés túl kevés, a dupla soremelés viszont túl sok. Ezért programunk minden soremelés karakter hatására egy normál és egy grafikus sort dob. (A normál sor 1/6 inch, a grafikus 1/9 inch.) A szokásos sornyomtató papír 12 inch hosszú, erre pontosan 72 normál sor fér. Normál + grafikus sorból nem lehet kirakni a 72 inchet, 43 db fér a papírra, de még marad 1/18 inch. Ha azonban a sorok száma 42 vagy kevesebb, a maradék mindig kirakható valamennyi grafikus és valamennyi normál sorból.

A lapdobás

A felhasznált nyomtatónak nincs lapdobás funkciója, ezt üres sorok kiírásával kell szimulálni. Az Easy Script szövegszerkesztő tudja ezt, lapokra tudja tagolni a szöveget; de mivel megváltoztatjuk a sorközt, nekünk kell vezérelni a lapemelést is. Az Easy Script szerencsére a kimeneti fájlba nemcsak a beállított laphossznak megfelelő számú CR (ASCII 13) kódot írja ki, hanem egy nulla bájtsorozattal is jelzi a lap végét; ezt kell figyelni.

Megjegyzés: az Easy Script a „formázott" szöveget nyomtatóra és a képernyőre tudja kiírni. (A formázás az oldalakra osztást, oldalszámozást, margókiegyenlítést stb. jelenti.) Azonban – bár ez a leírásban nem szerepel – fájlba is ki tudja írni a feldolgozott szöveget. Ez az F1 O S utasításokkal érhető el. Amíg ezt nem tudtuk, becsaptuk az Easy Scriptet, hogy a lemezt nyomtatónak érezze. Indítása előtt kikapcsoltuk a (4-es egységszámú) nyomtatót, majd a (8-as egységszámú) hajlékonylemezt a használati utasításában leírt paranccsal átdefiniáltuk 4-esnek, és megnyitottunk benne egy szekvenciális fájlt. Ezután elindítottuk az Easy Scriptet. Igaz, a végén a lemezfájl lezáratlan maradt, de a directory bejegyzés átírásával le lehetett zárni.

MÁRK GÉZA

COMMODORE 64

Nyomtatás

Sokszor szükség van arra, hogy a képernyő tartalmát vagy annak egy részét kiírassuk a nyomtatóval. A következő szubrutin segítségével a képernyő bármely részén kijelölt, tetszőleges nagyságú téglalap kinyomtatható.

A bemenő paraméterek a következők:

S – hányadik képernyősortól kezdődjön a kiírás (1–25)
O – hányadik képernyőoszloptól kezdődjön a kiírás (1–40)
SH – hány karaktert íratunk ki egy sorból (1–40)
OH – hány sort íratunk ki (1–25)
PO% – kezdő pozíció a nyomtatón
NA$ – "N" – normál betűk
"I" – dupla szélességű betűk

```
10 REM ****************************
20 REM *
30 REM *   F O P R O G R A M
40 REM *
50 REM ****************************
60 S=1:O=1:SH=40:OH=25:NA$="N":PO%=1
100 GOSUB 5000
200 STOP
300 END
5000 REM ****************************
5010 REM *
5020 REM *  H A R D C O P Y
5030 REM *
5040 REM ****************************
5050 IF S<1 OR S>25 THEN 5100
5060 IF O<1 OR O>40 THEN 5100
5070 IF SH<1 OR SH>40 THEN 5100
5080 IF OH<1 OR OH>25 THEN 5100
5082 IF PO%<1 OR PO%>80 THEN 5100
5084 IF (PO%+SH)>81 THEN 5100
5086 IF NA$="I" AND PO%<>1 THEN PZ%=PO%/2
     :IF 2*(PZ%+SH)>80 THEN 5100
5090 IF (S+OH)<=26 AND (O+SH)<=41 THEN 5125
5100 OPEN10,4
5110 PRINT#10:PRINT#10:PRINT#10,
     "  H I B A S   P A R A M E T E R E Z E S  ! !"
5120 CLOSE10:RETURN
5125 N$=CHR$(14):A=1023+(S-1)*40
5130 OPEN10,4:PRINT#10:PRINT#10:CLOSE10
5140 FOR I1=S TO S+OH-1
5150 SO$=""
5160 FOR I2=O TO O+SH-1
5170 B=PEEK(A+I2)
5173 IF B>127 THEN B=B-128
5176 IF B>31 AND B<64 THEN 5190
5180 IF B<32 THEN B=B+64:GOTO 5190
5183 IF B>63 AND B<96 THEN B=B+32:GOTO 5190
5185 IF B=96 THEN B=32:GOTO 5190
5187 B=B+64
5190 SO$=SO$+CHR$(B)
5200 NEXT I2
5202 IF PO%=1 THEN 5210
5204 IF NA$="N" THEN PZ%=PO%-1
5206 FOR I3=1 TO PZ%:SO$=" "+SO$:NEXT I3
5210 OPEN10,4
5220 IF NA$="I" THEN PRINT#10,N$SO$:GOTO5240
5230 PRINT#10,SO$
5240 CLOSE10
5250 A=A+40
5260 !EXTI1
5270 OPEN10,4:PRINT#10,CHR$(15):CLOSE10
5280 RETURN
```

Mint ahogy a paraméterek ismertetésénél már kiderült, meg lehet adni, hogy a nyomtató hányadik pozíciójától kezdődjön a kiíratás. Továbbá ha grafikus nyomtatónk van, választani lehet normál és dupla szélességű betűkészlet között. A szubrutinnal elsősorban tablók tervezhetők a képernyőn, nyomtatás előtt, de bármilyen másik programban jól használható. Ha például behívását egy funkciógombhoz kötjük, akkor gombnyomásra az eredmények nemcsak a képernyőn jelennek meg, hanem a nyomtatón is.

RICHTER JÖRG

Nyomvonalas létesítmények hossz- és keresztszelvényeinek szerkesztése és rajzolása

A beruházás előkészítés és tervezés egyik igen fáradságos és munkaigényes fázisa a terepadatok megmérésére, számítása, szerkesztése és rajzolása. Ebből a folyamatból az utóbbi három elemet könnyíti meg a program alkalmazása.
Az input a geodézia felmérési jegyzőkönyv adatai. Nyomvonalas létesítményeknél ezek a szelvényszám, a látósík, a részletpont távolsága a felmérési alapvonaltól, valamint a magassági lécolvasás értékei. Az adatbevitel – feldolgozás során önállóan is végezhető – egyszerűsített formában az állandóan előforduló karakterek nélkül történik.
A részletpont magasságok számítása után az ábrázolás a terepvonalon túlmenően tartalmazza a tereptárgyak méretarányos jelét, a szelvényszámot és a magassági lépcsőt is.
A hossz-szelvény szerkesztéséhez összeválogatja az ide tartozó mérési adatokat, majd a kívánt léptékű rajzot állítja elő.
Különösen jól használható meglévő nyomvonalas létesítmények felmérésénél, valamint tervezett nyomvonalas létesítmények (út, csővezeték) terepfelvételénél.

Meliorációs munkák költségvetés-készítése

A mezőgazdaság termelőképessége fokozásának, illetve a termőföld és a környezetük megvédésének fontos eleme az érintett területen a meliorációs (vízrendezési, területrendezési stb.) munkák elvégzése. Ezen tervezési folyamat befejező munkarésze az építési munkák várható költségeinek kalkulálása.
Az anyagok és keverékek munkahelyi anyagárának ismeretében az adatbázisban szereplő Építőipari Költségszámítási Normák (ÉkN) tételei adattartalmával, valamint az egyes tételekhez tartozó mennyiségekkel számítja az anyagköltség és díj értékeket építményenként (Megj.: a meliorációs munka sajátossága, hogy egy-egy tételnél 1–50 építmény is szerepelhet). A tételek feldolgozása után az egyes építményekre külön-külön építményösszesítőt készít az érvényes árképzési jogszabályok szerint. Ezt követően ugyanilyen formán készül a beruházás főösszesítője.
Az output formátuma a költségvetés-készítési előírásoknak megfelelő, közvetlen sokszorosításra alkalmas kivitelű.
Az adatbázis az általában előforduló ÉkN tételeket tartalmazza, de igény szerint bővíthető akár „M", akár „K" tétel is. Az adattár karbantartás során az esetleg változó adatok (pl. díj) gyorsan módosíthatók.
A legfontosabb adatok archiválásra kerülnek, hogy később is hozzáférhetők legyenek.

Keverékszámítás (3-as ÉkN)

A tervezési folyamat lényeges része a beruházás várható költségeinek tervezése a műszaki dokumentáció alapján. Ennek rész munkafolyamata az építéshez felhasználásra kerülő anyagok, keverékek árának alakulása.
A munkahelyi anyagárak (depónia árak), valamint a szállítás távolsága ismeretében az adatbázisban tárolt keverék összetevő normákból, valamint az elkészítés díjából az érvényes árképzési jogszabályok szerint kerül számításra a keverékár.

Hardverigénye: C 610
mágneslemez meghajtó (1 db)
rajzgép (WATANABE MP 1000)

Hardverigénye: C 610
mágneslemez meghajtó
nyomtató

Commodore 610 felhasználók figyelmébe ajánljuk az alábbi programcsomagokat

Az adatbázis az Építőipari Költségszámítási Normák (ÉkN) 3. kötetének – Épületek és műtárgyak felépítményi szerkezetei – valamennyi keverékét tartalmazza.
Ez a díjak tekintetében árhatósági intézkedésre módosítható.
Vállalati, egyedi adatokkal bővíthető.
Az output formátuma az árképzési előírásoknak megfelelő, címében az adott keverék ÉkN és a vonatkozó műszaki szabvány szerinti megnevezése is szerepel.

Hardverigénye: C 610 mágneslemez meghajtó nyomtató.

Szerződések nyilvántartása

A program tárolja a tervező, beruházási vállalat szerződésállományát, illetve annak legfontosabb adatait, úgymint munkaszám, megrendelő neve, megrendelés tárgya, díja, határideje, a szerződés állapota (ajánlat, érvényes, lemondott stb.), valamint belső szervezeti megbontása.
A feldolgozás során mód nyílik új adatok bevitelére, az adattartalom korrigálására, valamint különböző feltételek szerinti lekérdezésre.
A kigyűjtés történhet:
- munkaszám
- a beruházó neve
- a beruházó gazdasági szektora
- díj
- határidő
- szerződés állapota
- belső megbontása

illetve ezek egymásra épülő kombinációja szerint (pl. a beruházó neve és a szerződés állapota és határideje és stb. szerint).
A kiválogatott adatállomány nyomtatása történhet:
- betűrend
- gazdasági szektor
- határidő sorrendje
- szerződés állapota
- termelési egységek

szerinti csoportosításban.
Az állomány naprakész volta biztosítja bármely percben a különböző vezetési szintek információ igényét a munkaellátottságról. Azonnal nyerhetők olyan adatok, melyek elemzésére korábban információ híján nem került sor (pl. lemondott munkák aránya).

Hardverigénye: C 610 mágneslemez meghajtó (1 db) nyomtató (132 karakteres)

Beruházási adatok elemzése

A beruházások tervezett élettartama alatti műszaki szempontú viselkedése mellett igen fontos, hogy azok az adott közgazdasági környezetben várhatóan milyen eredményt fognak létrehozni. Ennek az eddigieknél hathatósabb vizsgálatát könnyíti meg a program a dinamikus megtérülési idő és a várható eredmények gyors kiszámolásával.
Az elemzett építmény várható élettartama alatt tervezett beruházási és pótlási költségekből, a termelés költségeiből és eredményeiből, valamint a forgóeszköz-növekményekből és maradványértékből számított egyenleg alapján képezi a belső megtérülési rátát. Az egyenleg nettó jelenértékét előállítva számítja a dinamikus megtérülési idő értékét.
A fejlesztés nélküli, a fejlesztés utáni költség és termelési érték adatokból meghatározza a nettó eredménynövekedés, haszon költség arány és az évenként átlagosan elérhető eredmény értékeit.
A program inputja megfelel a Világbanki hitelből finanszírozott beruházások közgazdasági elemzésénél használatos adatsornak. Outputja dokumentálható formában adja a kiindulási és az eredmény adatokat.

Hardverigénye: C 610 mágneslemez meghajtó nyomtató.

Építsünk számítógépet! IX.

Mindenekelőtt néhány hír a gépépítésről. Megismétlem, hogy a kártyák megrendelhetők. Vidéki csoport alakulásáról nem tudok, de a pestibe jár néhány Budapest környéki érdeklődő is. Ígéretet kaptunk arra, hogy októbertől lesz egy felműszerezett kis műhelyünk. Összejöveteleink helye változatlanul a TIT Stúdió (Bp. XI., Bocskai út 37.), minden páros hét csütörtökén, 6–8 óra között.

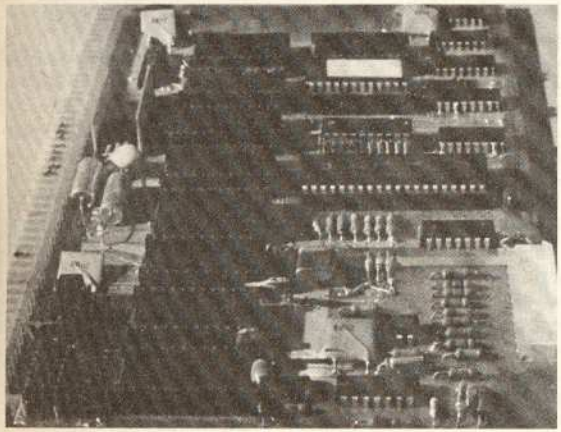
Úgy, mint máskor, a cikk írására csak néhány napot hagytam magamnak, hiszen egy iratgyűjtőben együtt volt minden, amit csak írtunk, rajzoltunk erről a kártyáról a fejlesztés, módosítás során. Előre kikészítettem a „6801" feliratú gyűjtőt az asztalomra, de mikor írni akartam a cikket, nem volt sehol. Hozzám sokan jönnek, és valószínűleg valaki magával vitte. Csak erre gondolhatok, mert saját jegyzeteimet én még sohasem adtam oda másnak, hiszen „szép" írásom miatt el sem tudná olvasni. Így most abba a furcsa helyzetbe kerültem, hogy a saját kártyámat kell „lekoppintanom", amihez természetesen sokkal több idő kell, mint egy egyszerű cikk megírásához. Ezért ezúttal a kártya kisebb részéről szólok, mint ahogy terveztem.

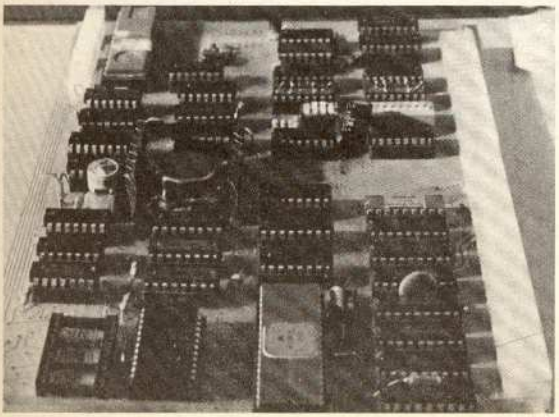
Elkezdem tehát az alapkiépítésű gép utolsó és legfontosabb kártyájának, a központiegységkártyának ismertetését. Az elnevezés pontatlan, mert nemhogy a kártya, de még a 6803 típusú IC is ellát a központi egységtől elvárt funkciókon felülieket is. Mit csinál egy központi egység? A gép alapműködésének program szerinti vezérlését, információcserét a többi egységgel. Ezen belül beolvassa a máshol tárolt program soron következő részletét (egy utasítás), azt a feldolgozásig tárolja, értelmezi és végrehajtja, figyeli és az előre programozott módon és időben végrehajtja a más egységektől érkező üzemmód-módosítás kérést. Amit ezenkívül tesz, azt a mikroprocesszor ismertetésénél már elmondtam.

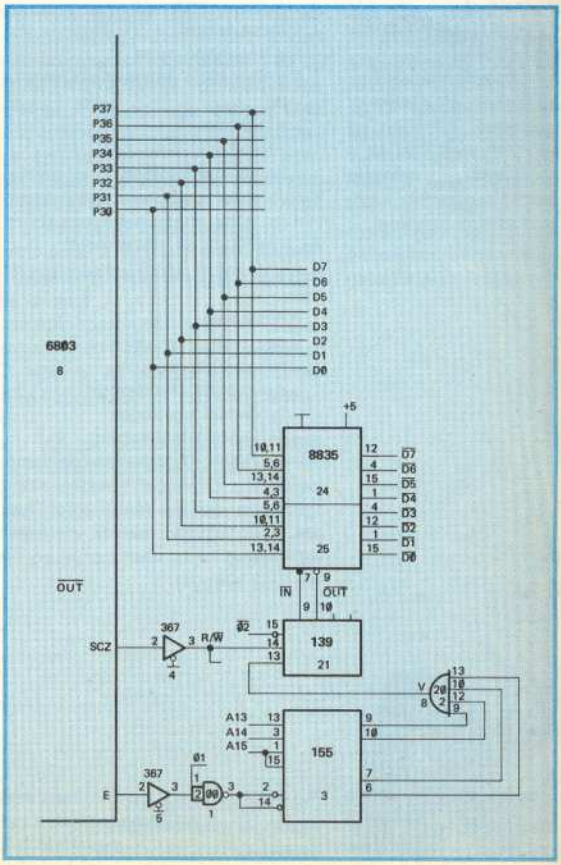
Milyen funkciók valósulnak meg a kártyán? Ez a kártya gyakorlatilag egy mikroszámítógép, mert ha megkapja a szükséges tápfeszültségeket, amiket a busz leírásánál ismertettem, közvetlenül összekapcsolható magnetofonnal, billentyűzettel vagy terminállal stb. A kártyán 2–16 k ROM és 2–16 k RAM számára van hely és címdekódolás. A kártya vezérli a busz adatforgalmát, címvezetékeit, az írás/olvasást, megszakításkezelést, a kártyák együttműködését. Ezenkívül egy 8 bites párhuzamos adatcsatornán és egy kétirányú soros csatornán közvetlen kapcsolatot tart a külvilággal. Ez utóbbiak a billentyűzet (A/D átalakító, magnetofon, nyomtató, kapcsolós botkormányok) és a magnetofon (RS232 soros vonal) közvetlen használatát teszik lehetővé.

Az eddig megszokottakkal megegyezően közlöm a kártya két oldaláról készített *fényképeket*. (Több olvasó kifogásolta a képek minőségét. Valóban, az utolsó kártyáról megjelent fénykép alig felismerhető, de az alapkártyáé olyan jó volt, hogy annak alapján két középiskolás elkészítette a NYÁK-tervet.)

A mikroprocesszor az adat-, cím- és vezérlővonalakon keresztül kapcsolódik a buszra. Az adatok adás/vétel vezérlő része az *ábrán* látha-







tó. Az adatvonalak vezérlését az SC2, E, P45–P47 mikroprocesszor-jelek végzik. A mikroprocesszor (IC8) harmadik párhuzamos csatornája (P30–P37 vezetékek, ill. 37.–30. láb), mint multiplexelt adat- és címcsatorna működik. Adatcsatorna állapotban a D0–D7 adatvezetékeken keresztül adatforgalmat bonyolít le, részben a kártyán levő memóriákkal, részben a busszal. Ez utóbbi a két invertáló buszhajtón (IC 24, 25) keresztül történik. A buszhajtók az adatokat a kiválasztott irányba engedik tovább. A kiválasztást a 7. lábakra adott IN, vagy a 9. lábakra adott OUT jel teszi.

E két iránykijelölő jel beállítását egy 2-ről 4-re dekódoló/demultiplexer (IC 21) egy része végzi a Ø2, R/W és a V jel alapján. A buszra közvetlenül kikerülő R/W jelet a mikroprocesszor SC2 jeléből egy buszhajtó (IC 4) egy része állítja elő. A buszra ugyancsak kikerülő Ø2 jelet egy kétbemenetű ÉS–NEM kapu (IC 1) egy része hozza létre a Ø1-inverz órajelből, melyet egy buszhajtó (IC 5) egy része állít elő a mikroprocesszor E órajeléből.

Az órajel és az írás/olvasás jel egyaránt azonos értékű akár a busz, akár a kártyán levő memória az adatmozgás egyik végállomása. A V jel szabja meg, hogy a mikroprocesszor a busszal vagy a kártyán levő memóriával lépjen kapcsolatba. Ezt a jelet egy négybemenetű ÉS–NEM kapu (IC 2) egy része állítja elő.

Ezt a négy bemenetet egy kapuzott, három vonalról nyolc vonalra dekódolóként működő, 2-ről 4-re dekódoló/demultiplexer (IC 3) hozza létre, az órajelből és a három legmagasabb helyiértékű címből (A13–A15). A kártyán a 4000–7FFF és C000–FFFF címterületet használja a memória (valamint a mikroprocesszor belső memóriája, amelyet a mikroprocesszor belsőleg vezérel, a 00–1F és 80–FF címterületet). Így a busszal dolgozik együtt a mikroprocesszor, ha 0000–3FFF vagy 8000–BFFF címterület van kiválasztva, vagyis A14=0.

Megvizsgálva a 74LS155 működési táblázatát, látható, hogy az A és a G21, G22 bemenetek – esetünkben az A13 és A15 címvezeték – értékétől függetlenül, csak a B bemenet – esetünkben az A14 címvezeték – határozza meg, hogy mikor van a Q02, Q12, Q01, Q11 kimeneteken – esetünkben az IC 2 bemenetei – magas jel egyidejűleg, ami a V jel alacsony értékét adja. Ez az A14=1 eset.

A kártya további részeiről a következő folytatásban lesz szó. DR. SIMONYI ENDRE

BÖRZE

Szoftver-börzét rendez a Neumann János Számítógéptudományi Társaság és a Mátra Művelődési Központ Gyöngyösön (Nemecz J. tér 3.), 1985. szeptember 14-én, szombaton 10–18 óráig. A rendezvény védnöke Páris György, a Tudományszervezési és Informatikai Intézet igazgatója.

A börzén bárki bemutathatja, elcserélheti programjait. Ehhez helyszíni gyorsmásolás is igénybe vehető. A rendezők elsősorban a személyszámítógép-tulajdonosokat, a mikroklubok és az NJSZT tagjait várják.

A szoftver-börzén az érdeklődők megismerhetik számos számítógép- és szoftverfejlesztéssel foglalkozó hazai vállalat, kutatóintézet, munkaközösség eredményeit. A rendező szervek előzetes piackutatása alapján lehetőség nyílik a gyártók és a Heves megyei üzemek, termelőszövetkezetek, intézmények üzletkötésére is.

TUDÁSPRÓBA

1. Az alfabetikus jelek betűkből, azaz az ábécé (idegen szóval alfabéta) elemeiből állnak, a numerikus jelek pedig számokból (numerusokból). Igaz-e, hogy az alfanumerikus jelek betűkből és számokból épülnek fel?

2. Ellentétes-e egymással az abakusszal (golyós számológéppel) való számolás és az algoritmus szerinti (más szóval algoritmikus) számolás, amelynek megjelenését egyesek al-Chvarizmivel hozzák kapcsolatba?

3. Igaz-e, hogy „a szögfüggvények periodikusak"?

4. Mi a különbség törtvonal és töröttvonal között?

5. Mi a PC?

6. Melyik a nagyobb szám, a 2 alapú számábrázolási rendszerben a 10101010101010101 vagy a 16 alapú számábrázolási rendszerben az A0A0A?

7. Mi az ellenőrző szám?

8. Van-e olyan (alapszámos, helyiértékes) számábrázolási rendszer, amelyben nincs az előjel számára hely fenntartva?

9. Van-e olyan helyiértékes számábrázolási rendszer, amelyben az összeadásnál az átvitel nem jobbról balra, hanem balról jobbra történik?

10. Mi a mix és mire használják?

11. Van-e kristályos folyadék- („krifó") kijelzős táskaszámológép a piacon?

12. Mi a PIA?

13. Mi a posztprocesszor?

14. Mi a disc drive?

a) lemezmeghajtó, b) lemezjátszó, c) tárcsa lejátszó, d) az, amit a disc jockey csinál, e) lemez író-olvasó egység, f) lemezegység

15. Minek kell eszébe jutnia egy számológéptudományban jártas személynek, ha a Pascal szót hallja?

16. Mi a CAD?

17. Mi a CAM?

18. Mi következik abból, hogy „A DNS, a genetika, a génsebészet, a hibridkukorica szintén tipikusan informatikai problémakör, és már ma is lényeges összetevői népgazdaságunknak, életszínvonalunknak, külkereskedelmünknek."

19. Mi a különbség információátvitel és információátmásolás között?

20. Mi a flag?

Összeállította: TAKÁCSY ILDIKÓ

ADOK – VESZEK – CSERÉLEK

Ebben a rovatunkban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás közleteknek épelt soronként (60 karakter) 100,– Ft, magánszemélyeknek az első sor 50,– Ft, minden további sor 20,– Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

- **ZX–81 PROGRAMOKAT** cserélek. Rada Illés, Jászberény, Tompa Mihály u. 1. 5100
- **SANYO PHC–25** tulajdonosokat keresek tapasztalatcsere és programcsere céljából. Makáry György, Kecskemét, Sárkantyú u. 3. 6000
- **HÉT DB TI99/4A** számítógép, magnóval, bővített BASIC-kel, rengeteg programmal, szakkönyvekkel darabonként is nagyon olcsón eladó. 16 bites processzor, 16 kbájt RAM, 62 kbájt ROM, 16 szín, hang- és zajgenerátor. Megbízható, strapabíró írógép-tasztatúra. Iskoláknak, kluboknak ideális. Telefon: 497-131 és 211-214
- **COMMODORE 64-ES** programokat cserélek. Csíkos Gabriella, Pécs, Szalai u. 12/B. 7622
- **ELADÓ ALIG HASZNÁLT ZX–81** videokimenettel és 64 k memóriával, külön-külön is. Kilián János, Szekszárd, Babits Mihály u. 17. 7100. Telefon: 74/11-817 (munkaidőben).
- **COMMODORE VC20-AS,** C64-es, C116-os, C16-os, C264-es (plusz 4) mikroszámítógépekhez szoftvereket cserére, különböző kiegészítő berendezéseket pedig eladásra kínálunk. VC20-hoz grafika, hang stb. kezelési tájékoztatók, C16-hoz (C116-hoz) pedig magyar nyelvű kézikönyv, demo kazettával rendelhető munkaközösségünktől. Kérje díjtalan tájékoztatónkat! INTER GM. Kaposvár, Rákóczi tér 1. 7400
- **ZX-SPECTRUM** 48 k + interfész + microdrive 30 ezer forintért eladó. Ingyen adok még 3 db microdrive kazettát, különböző programokat (lineáris programozás, full compiler stb.) és játékprogramokat (sí, sakk, repülés, űrjátékok). Balogh András, Bp. Kokárda u. 25. 1028. Telefon: 155-865
- **C64 SZAKTANÁCSADÁS,** kész (esetleg nyugati) rendszerek, adatbázisok illesztése, testreszabása, igény szerint betanítással. INFO GMK, Budaörs, Patkó u. 13. 2040
- **SZÍNVONALAS, SOKOLDALÚ,** szórakoztató matematikai oktatóprogram kisiskolásoknak, 48 k-s Spectrumra. Ára: 200,– Ft. Csak akkor fizet, ha megnyerte tetszését! Mitró István, Miskolc, Dessewffy u. 4. 3529
- **SZERETNÉK KAPCSOLATBA KERÜLNI** más Sinclair ZX-Spectrum tulajdonosokkal, programcsere céljából. Sokvári Attila, Szolnok, Jászi Ferenc út 12. 5000
- **PRIMÓHOZ ÍRT FORTH** (16 kbájt, száznál több utasítás) után érdeklődni lehet: Szabó Imre, Pécs, Péter u. 1. 7625 Telefon: 72/12-660
- **FELAJÁNLOM AZ ÉRDEKLŐDŐKNEK** az alábbi nyugatnémet újságokat: Microcomputer Zeitschrift, 1983. és 1984. évi 1–12. szám, CHIP, 1983. 1–12. szám, CT Magazin für Computer Technik, 1984. 1–12. szám. A teljes sorozatok ára összesen 3000 Ft (az 1984. évi sorozatoké 1000,– Ft, az 1983. évieké 500,– Ft). A kívánt címre az újságokat utánvéttel postafordultával elküldöm. Németh Józsefné, Győr, Alkotmány u. 20. 9021
- **COMMODORE 64 ALAPGÉP** és 1541-es meghajtó eladó, esetleg hosszabb időre bérbe vehető. Érdeklődés a 487-277 telefonon 17 óra után.

A Metalloglobus Vállalatnál 6 db 2 × 2,5 Mbájtos CM 5400-as bolgár cserélhető lemezes diszkegység újszerű állapotban eladó.

Érdeklődni lehet:
Martin József
vagy Bozók József

Budapest X., Jászberényi u. 57.
Telefon:
277-085 vagy 276-816

Kazetta-szabvány

Legutóbb az ún. Kansas City-szabvány kialakulásáról, jellemzőiről írtam. Ismertettem előnyeit, hátrányait és egy olvasóprogramot. Most a kazettára író programról lesz szó. Ez előtt azonban még visszatérek az előző cikk néhány megállapítására.

A Kansas City-szabvány hátránya a kis sebesség. A 300 bit/s átviteli sebesség a szokásos start-stop bit miatt bájtonként 10 bitet jelent. Ez az ASCII-kódok használata esetén megkétszereződik, végül a fejlécz, a bájtszám, a kezdőcím és az ellenőrzőszám megadása azt eredményezi, hogy minden egyes bájt 25 bit átvitelét jelenti. Másképpen fogalmazva: 12,5 bájt/s az átviteli sebesség! Így egy 8 k-s BASIC fordító betöltése kb. 11 perc.

Kezdettől fogva igyekeztek növelni a sebességet, az előnyös tulajdonságok (egyszerű hardver, szoftver, olcsó magnetofon, szalag) megtartása mellett. A gyorsítási módszerek két irányban haladtak. Az egyik a bájtonként átvitt bitek számát igyekezett csökkenteni. Ennek érdekében nem hexadecimálisan kódolt ASCII-kódokat használtak, hanem közvetlen bináris kódolást. Ez kétszeres hatékonyságot, de kisebb megbízhatóságot adott. Az átvitt sorok megnyújtása kevésbé növelte a hatékonyságot, de ez is kisebb megbízhatóságot adott. Mindezek együtt lehetővé tették azt, hogy éveken keresztül dolgozhattam ugyanazzal a magnetofonnal és kazettával, mintegy 30 bájt/s átviteli sebesség mellett.

A másik irány a jelfrekvencia növelésével, illetve a ciklusok számának csökkentésével növelte az információátvitel sebességét. A frekvencia megkétszerezése (2400 és 4800 Hz értékre) még a legtöbb magnetofonnál és szalagnál megengedhető volt. A megnégyszerezés azonban már sztereodeck és jó minőségű szalag használatát követelte meg. Az eredetileg négy és nyolc ciklus (0 és 1) felére, sőt negyedére csökkentése sem okozott problémákat. Szintén évek óta használom azt a változatot, amelynél a cikusszám egy és kettő, a frekvencia 2400 és 4800 Hz, az átvitel bináris kódolású, és a teljes programhoz egyetlen fejrész és ellenőrzőszám tartozik. Így az átvitel sebessége 240 bájt/s.

Az előző számban volt három programlista is. Egy sajnálatos tévedés miatt (az első papírra vetett, ki nem próbált változat listáját adtam le) a harmadik listában több hiba van. (A sors iróniája, hogy ezt megelőzően velem hasonló még nem fordult elő, és éppen most, amikor megtörtént, írok a szaklapokban megjelenő programok hibáiról.) A hibák és a helyes értékek a következők:

A 80-as sor elejére betoldandó: B=0:

A 145-ös sorban helyesen C1=C

A 200-as sorban 4 helyett mindenütt 6 áll, valamint betoldandó Y után: ,A1

Az íróprogram, az olvasóhoz hasonlóan, DRAGON BASIC-ben készült. Az *1. listában* látható töltő és szalagra kiíratő program olyan utasítást nem tartalmaz, ami az előző cikkben leírtakat is figyelembe véve, ne lenne érthető. A program a szalagra fejléc nélkül, start és stop bittel ellátott, binárisan kódolt bájtsorozatként viszi ki a kezdő- és végcímmel megadott tárolóterületet. A kezdő- és végcímet a szalagra nem írja rá. A bevezető rész a szalagon 1792 db "1" jel, ami kb. 6 s ideig tart. Ez bőségesen *elég a* magnetofon felgyorsulásán kívül a szinkronozáshoz is.

A *2. lista* itt is 6809 gépi kódú. Megértéséhez és más mikroprocesszorra átírásához részletes magyarázat található a negyedik oszlopban. A frekvencia beállítása a jelzett sorokban történik. Itt lehet a frekvenciamódosításhoz szükséges állandóértéket is beállítani. Ugyancsak jeleztem a ciklusszám-állandók helyét, az esetleges módosítás okából. Ezekkel óvatosan bánjunk, mert csak az olvasóprogramunkkal azonos frekvenciával és ciklusszámmal dolgozó íróprogram által készített szalag olvasható!

Az olvasóprogram alapján tetszés szerinti fejlécű, ellenőrzőszámú, tárolási formájú íróprogramot lehet készíteni. Szívesen adunk helyt a más gépekre, mikroprocesszorokra készült változatoknak is.

DR. SIMONYI ENDRE

Címek: K(ezdőcím): 3800–1
V(égcím): 3802–3
T1: 3804
T2: 3805
PIA-cím: FF20

Program:

```
8D 19     START  BSR FEJ       Négyszög-
                                hullám-képzés
86 02            LDA#2        a szükséges frekvenciával
BA FF20          ORA PIA
B7 FF20          STA PIA
BE 3800          LDX K
30 82            LEAX ,_X     Kezdőcím-beállítás

A6 80     2      LDA ,X+      Olvass be új adatot
BC 3802          CMPX V       Van még adat?
26 01            BNE I        Igen
39               RTS          Vége
8D 19     I      BSR IR       Tegyél ki egy adatot
20 F4            BRA 2
7F FF21   FEJ    CLR PIA+1    PIA-inicializálás
86 02            LDA#2        Adatirány-beállítás
B7 FF20          STA PIA
86 04            LDA#4        Port-engedélyezés
B7 FF21          STA PIA+1
BE 06FF          LDX#06FF     "1" sorozat kiírás
8D 34     1      BSR IR1
30 82            LEAX ,_X

26 FA            BNE 1
39               RTS
34 06     IR     PSHS B,A     Egy adatbájt start
                                és stop bittel

B7 3804          STA T1
C6 08            LDB#8
8D 1C            BSR IR2      Állítsd elő a start bitet
B6 3804   IR3    LDA T1       Válaszd le a legalsó bitet
84 01            ANDA#1
26 04            BNE IR4      Tedd ki
8D 13            BSR LR2      "0" adatbit
20 02            BRA IR5
8D 19     IR4    BSR IR1      "1" adatbit
B6 3804   IR5    LDA T1       Egészítsd ki
                                a szükségesekkel

47               ASRA
B7 3804          STA T1
5A               DECB         Folytasd az utolsó bittig
26 E9            BNE IR3
8D 0D            BSR IR1      Állítsd elő a stop bitet
35 04            PULS B       Töltsd vissza a B-regisztert
39               RTS          Térj vissza
34 06     IR2    PSHS B,A     "0" kiadáshoz tedd el
                                a B,A-t
C6 04            LDB#$4       1200 Hz-es állandó
86 48            LDA#$48      frekvencia beállító
20 06            BRA IR6
34 06     IR1    PSHS B,A     "1" kiadáshoz tedd el
                                a B,A-t
C6 08            LDB#8        2400 Hz-es állandó
86 21            LDA#$21      frekvencia beállító
B7 3805   IR6    STA T2       Késleltetés-beállítás
86 02     IR7    LDA#2        Négyszöghullám-képzés
BA FF20          ORA PIA      a szükséges frekvenciával
B7 FF20          STA PIA
B6 3804          LDA T1
4A        IR8    DECA         Késleltetés
26 FD            BNE IR8
86 02            LDA#2        Invertáld a hullámot
B8 FF20          EORA PIA
B7 FF20          STA PIA
B6 3804          LDA T1
4A        IR9    DECA         Késleltetés
26 FD            BNE IR9
5A               DECB         Ha nem kész,
26 E1            BNE IR7      folytasd
35 06            PULS B,A     B,A-regiszterek
                                visszatöltése
39               RTS          Térj vissza
```

2. lista

1. lista

```
5 REM IRO PROGRAM
10 CLS:CLEAR 500,&H37FF:D=&H3880:PRINT"A PROGRAM A 3800-3893 TERUELETET HASZNAL
JA!":PRINT"BEHELYEZENDOE A KAZETTA! A MAGNETOFONT IRAASRA KELL AALLITANI!"
20 FOR I=0 TO 137:READ A$:POKE D+10+I,VAL("&H"+A$):NEXT I
30 PRINT"KEZDOE EES VEEGCIM?":INPUT B,C:MOTORON:B1=INT(B/256):B2=B-(256*B1):C1=I
NT(C/256):C2=C-(256*C1):POKE D,B1:POKED+1,B2:POKED+2
,C1:POKED+3,C2:EXEC D+10:STOP
40 DATA 8D,19,86,02,BA,FF,20,B7,FF,20,BE,38,00,30,82,A6
50 DATA 80,BC,38,02,26,01,39,8D,19,20,F4,7F,FF,21,86,02
60 DATA B7,FF,20,86,04,B7,FF,21,8E,06,FF,8D,34,30,82,26
70 DATA FA,39,34,06,B7,38,04,C6,08,8D,1C,B6,38,04,84,01
80 DATA 26,04,8D,13,20,02,8D,19,B6,38,04,47,B7,38,04,5A
90 DATA 26,E9,8D,0D,35,04,39,34,06,C6,04,86,48,20,06,34
100 DATA 06,C6,08,86,21,B7,38,05,86,02,BA,FF,20,B7,FF,20
110 DATA B6,38,04,4A,26,FD,86,02,B8,FF,20,B7,FF,20,B6,38
120 DATA 04,4A,26,FD,5A,26,E1,35,06,39
```

T. SZERZŐ KOLLÉGÁK!

A honoráriumok zökkenőmentes átutalása érdekében kérjük, hogy az írásokkal, cikkekkel együtt az alábbi adatokat is szíveskedjenek szerkesztőségünkbe elküldeni:

Név
Születési hely, idő
Anyja neve
Lakáscím, telefonszám
Munkahely megnevezése
Munkahelyi címe, telefonszáma
Személyi száma
Utalás esetén melyik címre kéri a honoráriumot

Mit érnek a szaklapok?

Az utóbbi egy-két évben nagyon sok új szaklap jelent meg Nyugaton. Akinek lehetősége van rá, hogy néhányat rendszeresen kapjon, választás elé kerül: melyiket?

Minek alapján döntsünk? Az egyik szempont természetesen az, hogy „érdekes" cikkek legyenek benne, és ilyenből minél több. Milyen cikk „érdekes"? Ez érdeklődési körünktől függ. Egy gépépítőt hardvermegoldások, egy géphasználót programok érdekelnek elsősorban, míg egy leendő vásárlót az, hogy mit, hol, mennyiért lehet kapni.

A lapok egy része (például a BYTE, a CHIP) mindenkihez szól, mindenféle témáról közöl cikkeket, míg mások az olvasók valamelyik csoportjának írnak. Ez utóbbiak között vannak olyanok, amelyek egy bizonyos gép vagy gépcsalád összes témájával, mások valamelyik témacsoporttal, például a Happy Computer programokkal foglalkozik. Vannak még tovább szakosodottak is: így például a SOFTALK Apple programokról, a Dr. Dobb's Journal nagyobb rendszerek programjairól ír.

Feltételezve, hogy eljutottunk az érdekes cikkeket megjelentető lapokhoz, legtöbbször még mindig túl sok lap marad. Hogyan válasszunk ezek közül? A számunkra érdekes témák szerinti választás gyakran itt is segít, mert az egyik lap elsősorban, vagy kizárólag játékprogramokról ír, a másik valamilyen professzionális felhasználási területet vesz célba. Igen különös lap például a Hardcore Computist, mert ennek témája: hogyan kell másolni, felnyitni, módosítani másolás-védett programokat, melyeket Apple II-re írtak.

Amennyiben a lapban leírt megoldásokat, programokat használni akarjuk, olyan szempontokat is figyelembe kell vennünk, mint az ismertetés mélysége (van, akinek elég a feladat közlése, és már képes megoldani, de ez ritkaság; másoknak teljes részletességű használati utasításra és egész apró részletekre is kiterjedő ismertetésre van szükségük) és számos egyéb dolog. Véleményem szerint azonban legfontosabb a közöltek megbízhatósága.

Lapunk legutóbbi számában ismertettem egy minden géphez használható kazettaformátumszabványt, és az ennek megfelelően készített szalagok beolvasására szolgáló programot. A cikk egy 300 bit/s átviteli sebességű megoldást mutatott be, ami bár nagyon lassú, de lehetővé teszi programok, adatok átvitelét különböző típusú gépek között, speciális illesztőegység alkalmazása nélkül. Elgondolásom szerint egy második cikkben kívántam ismertetni a szalagra író programot.

Az első cikk elkészülte után azonban tudomást szereztem arról, hogy a c't nevű német nyelvű szaklap egy SUPERTAPE elnevezésű szalagformátumot ajánl hasonló célra, és sorozatban jelenteti meg az ismertebb gépekhez az író/olvasó programokat. Elolvasva a formátumot ismertető cikket, úgy véltem, hogy célszerűbb lenne azt használni. Ennek van ugyanis egy lényeges előnye, a nagy átviteli sebesség: 3600 bit/s, és egyes gépeknél ennek a kétszerese is lehet!

Miután megkaptam a C64 változatot, még inkább megtetszett, mert ez a másik változat (az első bizonyos hardvermódosításokat igényelt még) a Commodore BASIC lemezparancsainak használatát is lehetővé tette, készülékszámként a 7-et kijelölve. Azonnal beírtam a BASIC nyelvű töltő és működtető programot.

Sorszám	Hibás érték	Helyes érték
50	145751	145640
60	53000	52999
1250	105	104
1450	105	104
1440	50	49
1530	125	124
1540	125	124
1560	125	124
1570	125	124
1530	127	126
1550	127	126
1560	127	126
1570	127	126
1620	18	17
1690	201	200
1720	201	200

A programot lemezre vittem, majd ki akartam próbálni. Nem futott! Összeolvastuk többen, összesen hatszor, majd megállapítottam, hogy a beírás jó. A nem futás oka az volt, hogy a programban a betöltés hibátlanságának ellenőrzésére volt egy szám, a betöltendő kódok összege, amelytől eltérő kódösszeg esetén a program hibaüzenettel leállt. A számot kijavítva a talált kódösszegre, újrainditottam a programot. Nem futott! Mivel a töltőprogram BASIC része mindössze három sor volt, arról gyorsan meg lehetett állapítani, hogy hibátlan. Maradt a 78 soros DATA-tömbként megadott gépi kódú rész.

A betöltött programot szakaszonként assembler nyelvre fordítottam. Ennek vizsgálatát megkönnyítette a c't-ben megjelent assembler lista, amely azonban az egyes főbb szubrutinokat nem sorrendben, és a tárolócím nélkül közölte. Balszerencsémre a cikk egyik oldalát nem kaptam meg. Némi nyomozás után sikerült a szubrutinokat azonosítani, és meglepődve tapasztaltam, hogy összesen 18 JSR és JMP utasítás, valamint a kezdés címe hibás. A hibák nem adódhattak a programlistát készítő gép hibájából, mert minden cím a helyesnél eggyel nagyobb volt, és 15 helyen Ø helyett 1, 4 helyen 1 helyett Ø állt a legalacsonyabb helyiértékű bitnél. Ez utóbbiaknál viszont az eggyel magasabb helyiértéken 1 állt Ø helyett, sőt a kezdet címénél 1ØØØ állt Ø111 helyett.

Megállapítottam, hogy mind a 19 cím hibás a listában is. Tévedés vagy szándékosság? Nem tudom. Az azonban biztos, hogy a program nem használható! Követve az assembler listát látható volt, hogy elindítás után a program egy megszakításba – BRK – megy.

Ezek után két dolgot teszek: 1. mégis megjelentetem a „Kazettaszabvány" cikk második részét; 2. azok számára, akik szintén hozzájuttatok ehhez a programhoz, a *táblázatban* ismertetem az általam talált hibák helyét, a módosítandó és a helyes értékeket. A program teljes ellenőrzését még nem végeztem el. Az olvasók tapasztalatait is szívesen közöljük.

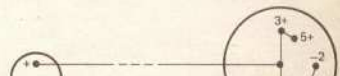
Nem futtatható vagy hibás eredményt szolgáltató programot, nem működő kapcsolást sok más lapban is találtam. Mi lehet ennek az oka? (A sajtóhibákkal, elírásokkal itt nem foglalkozom.) Részben az, hogy az olvasók által küldött anyagokat a lapok idő- és kapacitáshiány miatt nem ellenőrzik, és a szerzők egy része szándékosan hibásakat küld be. Az ismertetett példa nem ebbe a csoportba tartozik, mert a cikkben leírták, hogy a lap laboratóriumában azt tapasztalták, hogy bár egy közepes magnetofonnal, egyszerű illesztővel, jó minőségű szalaggal elérhető a 7200 bit/s, de Dataset használatával nem. A közölt listával viszont ilyen tapasztalatot nem lehetett szerezni.

Olvasóink választásának segítésére rendszeresen közölni fogjuk az ilyen negatív tapasztalatainkat, hogy felhívjuk a figyelmet a kétes megbízhatóságú cikkeket közlő lapokra. Ehhez az olvasók segítségét is kérjük.

DR. SIMONYI ENDRE

Programmentés magnetofonra

Valószínűleg sokakat bosszantott már, hogy Sharp PC–1211-ről vagy ZX81-ről MK–29-es BRG magnóra nem lehet eredményesen programot menteni, csak a legalább kétszer olyan drága japán Sanyo magnóra. Ezt az okozza, hogy a gépek kimeneti jelszintje mikrofonbemenetre van méretezve, így rádiómemeneten nem lehet rögzíteni. Mégis megoldható a probléma úgy, hogy a jack-dugasz negatív szálját a DIN-aljzat 2. pontjára, a pozitív szálat pedig elosztva az 1-es, 3-as, 4-es és 5-ös pontjára forrasztjuk. Ebbe az aljzatba kell az átjátszó drót egyik végét, a másikat pedig a magnó ötpólusú aljzatába dugni. Így csak a jack-dugasz helyét kell változtatni attól függően, hogy programot kimentünk vagy betöltünk-e.



A magnó hangerő-szabályozóját felvételkor a 6. és a 7. jel közé, hangszínszabályozóját pedig a 8. jelre kell állítani. Lejátszáskor szalagtól függően úgy kell beállítani a hangerő-szabályozót, hogy a 0. dB-es LED is világítson, a hangszínszabályozó pedig maradjon a 8. jelen.

LILÍNG ZOLTÁN

ZX–Spectrumra

KALAND-JÁTÉKOK

A játékprogramok egyik alaptípusa a kaland (angol szóval adventure) játék. A számkitaláló programok után a legrégibb fajta. Sok év telt el azóta, hogy az első ilyen, az ADVENTURE elkészült, PDP–10-re, azaz nagyszámítógépre.

A személyi számítógépek elterjedésével otthon is játszhatók lettek ezek a kalandozások.

A programok tárigénye miatt kezdetben kevés ilyen játék volt. Profi programozókra van szükség megírásukhoz. Nem a bonyolultság, hanem a házi számítógépek gyűszűnyi memóriája miatt. A nagyszámítógépekhez képest más lehetőségek bővebbek: a képernyőn nemcsak szöveg, hanem kísérő grafika is megjeleníthető.

Minden kalandjátékra jellemző a meseszerű alaptörténet és helyszín. A játékos sohasem lehet biztos előre semmiben – a játék velejárója a titokzatosság, ködösítés.

Példaként két, 1983-ban megjelent, Spectrumra írt programot mutatok be.

Az egyik legismertebb kalandjáték a HOBBIT. Ez szöveges játék, de a képernyő felső részén színes ábrák kísérik a történetet. A géppel való kommunikáció nyelve az Inglish/Inglish: a Melbourne Software bejegyzett márkaneve, az angol nyelv egy szűkített változata – BASIC angol a HOBBIT-hoz.

Meg lehet adni a mozgás irányát:

DOWN D, EAST E, NORTH N, NORTHEAST NE, NORTHWEST NW, SOUTH S, SOUTHEAST SE, SOUTHWEST SW, UP U, WEST W

Vannak speciális utasítások:

EXAMINE, HELP, INVENTORY I, LOAD, LOOK L, NOPRINT, PAUSE, PRINT, QUIT, SAVE, SCORE

Az akcióutasítások a következők:

BREAK, CLIMB, CLOSE, CROSS, DIG, DROP, DRINK, EMPTY, ENTER, EAT, FILL, FOLLOW, GIVE, GO, KILL, LOCK, PICK, PUT, OPEN, RUN, SAY, SHOOT, SWIM, TIE, TAKE, THROW, TURN, UNLOCK, UNTIE, WEAR

Prepozíciókat is tud használni a program:

ACROSS, AT, FROM, IN, INTO, ON, OUT, OFF, THROUGH, TO, UP, WITH

A határozószók:

CAREFULLY, GENTLY, QUICKLY, SOFTLY, VICIOUSLY

A program alaptörténete J. R. R. Tolkien könyvének, a Hobbit-nak meséje, amely magyarul „A babók" címen jelent meg. A programban a szereplők és a helyszín megegyezik a könyvbeliekkel: az orkok, a trollok, a Középföld, az Erdőelve. A játékos a könyv (a program) főszereplője: Bilbó, a hobbit. A játék célja: a Gyűrű megszerzése.

Más típusú kalandjáték az ATIC ATAC. Itt a játék nem szöveges parancsokkal folyik; akár egy arcade játéknál, billentyűzetről vagy botkormánnyal irányíthatjuk hősünket. A helyszín egy szellemkastély. A cél: megkeresni a trónterem kulcsának darabjait, és azokat összerakva, kinyitni a trónterem kapuját. Három játékmódozat van: a KNIGHT (lovag), a WIZARD (varázsló) és a SERF (paraszt).

A lovag a faliórákon, a varázsló a könyvszekrényeken, a paraszt a hordókon tud átmenni. A program méri az időt és az energiát. Ha az energia elfogy, oda egy ember.

A gyakorló változat a következő:

```
LOAD CODE : FOR I=24580 TO 24585 :
POKE I,Ø : NEXT I :
INPUT "HÁNY EMBER";X : POKE 32158,X :
POKE 23728, 233 :
RAND USR 24576
```

A segédprogram felhasználásával tetszőleges számú emberünk lehet. Itt is fel lehet venni tárgyakat: kulcsot, keresztet, üveget stb. A kulcsokkal természetesen a megfelelő ajtókat lehet kinyitni, a kereszt az ördögtől véd – mindegyiket fel lehet használni valamire.

Hogy ne legyen túl könnyű, a játékos egyszerre csak háromféle tárgyat tarthat magánál, és több, mint tízféle felvehető dolog van. A nehézséget fokozzák a mindenütt előtűnő szellemek is: a velük való találkozás jókora adag energiájába kerül az embernek.

Egyre több kalandprogram jelenik meg házi számítógépekre, hiszen az igény is egyre nagyobb. Ezekhez a játékokhoz ész is kell, és sokkal emberközelibbek, mint a harci játékok.

KOVÁCS BOTOND

ZX–Spectrumra

ŰRHAJÓ

```
1 BORDER 1
2 PAPER 1
3 CLS
4 INK 7
5 REM **********
6 REM *        *
7 REM * URHAJO *
8 REM *        *
9 REM **********
10 RANDOMIZE
20 LET P=0
30 LET O=4
50 LET X=15
60 LET Y=21
100 FOR I=1 TO 8
105 REM URHAJO
110 DATA 24,24,60,126,255,255,255,255
120 READ A
130 POKE USR "A"+I,A
140 NEXT I
150 LET B=INT (RND*10)
155 LET Z=(X*8)+12
165 REM JATEK VEGE
170 IF B>=29 THEN CLS:PRINT AT 0,0;"VESZTETT ";P;" PONTOT ERT EL"
PAUSE 0:RUN
172 REM URHAJO/URHAJO
175 PRINT AT O,B;"   O"
180 IF P<=90 THEN LET B=B+1
185 IF P>90 AND P<=490 THEN LET B=B+2
190 IF P>290 THEN LET B=B+3
330 REM IRANYITAS
340 IF X>=28 THEN LET X=28
350 IF X<=0 THEN LET X=0
355 PRINT AT Y,X;" A "
360 IF INKEY$="8" THEN LET X=X+1:GO TO 330
365 PRINT AT Y,X;" A "
370 IF INKEY$="5" THEN LET X=X-1:GO TO 330
375 REM LOVES
380 LET T=B+1: IF INKEY$="0" THEN PLOT Z,0:DRAW 0,140:PAUSE 10
INK 1:PLOT Z,0:DRAW 0,155:INK 7: IF T=X THEN LET P=P+10:BEEP .25,25
GO TO 1000
900 GO TO 155
950 REM ERTEKELES
1000 CLS : PRINT AT 0,0;P;" PONTOT ERT EL":PAUSE 0:GO TO 150
```

A program lényege, hogy kilőjük az alsó űrhajóval a fent közlekedő "O" betűt. Irányítás balra "5", jobbra "8", lövés "0"-val történik, ezért botkormánnyal is működik. A fent közlekedő űrhajó egy bizonyos pont elérése után felgyorsul: három fokozata van.

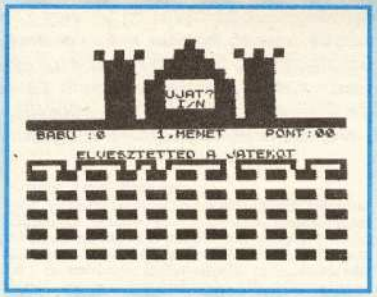
A változók: p = értékelés, x = jobbra-balra irányítás. A függőleges koordináták: az űrhajó 21-es, az "O" 5-ös. A 130-as, 355-ös és 365-ös sorba az "A" betűt a grafikus változatban kell beírni.

DÓZSA GERGELY

ZX81-re

KÖZÉPKOR-JÁTÉK I–II.

```
 1 GOSUB 1200
 2 PRINT "
 3 LET M=3
 4 LET P=0
 5 LET E=1
 6 LET T=13
 10 LET A$="
 15 LET B$="
 17 PRINT AT 9,1;"BABU :";M;AT
9,13;E;".MENET";AT 9,24;"PONT:";
P;AT
 20 PRINT AT 12,0;A$
 25 FOR I=13 TO 21 STEP 2
 30 PRINT AT I,0;B$
 35 NEXT I
 40 LET U$="
 42 LET Y=30
 45 LET O=30
 50 LET L=0
 55 LET K=11
 100 PRINT AT K,L;U$
 102 PRINT AT K,O;"
 104 LET O=O-1
 105 IF O=0 THEN GOTO 900
 106 IF L+1=0 OR L=5 THEN GOTO
110
 110 IF L=3 OR L=9 OR L=12 OR L=
19 OR L=26 THEN GOTO 500
 120 IF L=28 OR L=29 OR L=30 OR
L=31 THEN GOTO 800
 150 IF INKEY$="L" THEN LET L=L+
1
 155 IF INKEY$="H" THEN GOTO 100
0
 170 GOTO 100
 480 PRINT AT K,L+1;"  "
 490 LET K=K-1
 500 PRINT AT K,L+1;" "
 510 FOR I=0 TO 9
 520 PRINT AT K+1,L+1;"
 530 FOR F=0 TO 2
 540 NEXT F
 550 PRINT AT K+1,L+1;"
 560 FOR G=0 TO 2
 570 NEXT G
 580 GOTO 100
 800 PRINT AT K+1,L+1;" "
 806 LET H=M-1
 807 PRINT AT 9,7;M
 808 IF M=0 THEN GOTO 700
 590 LET L=0
 600 GOTO 500
 700 PRINT AT 11,5;"ELVESZTETTED
A JATEKOT"
 705 PRINT AT 5,14;"UJAT?";AT 6,
15;"I/N"
 710 LET D$=INKEY$
 711 IF D$="I" THEN RUN
 712 IF D$="N" THEN STOP
 713 GOTO 710
 800 LET E=E+1
 801 IF E=10 THEN LET T=12
 803 IF E=11 THEN GOTO 859
 805 PRINT AT K,L+1;" "
 810 PRINT AT 0,T;E
 820 LET P=P+10
 830 PRINT AT 9,29;P
 840 LET L=0
 850 GOTO 100
 860 PRINT AT 11,5;"MEGNYERTED A
JATEKOT."
 865 GOTO 705
 900 LET O=30
 910 PRINT AT 11,1;" "
 920 GOTO 100
1000 PRINT AT K,L+1;" "
1010 LET L=L+1
1020 LET K=K-1
1027 PRINT AT 11,0;"
1030 PRINT AT K,L;U$
1040 PRINT AT K,L+1;" "
1045 LET O=O-1
1046 IF O=0 THEN GOSUB 1000
1050 LET L=L+2
1060 LET K=K+1
1070 GOTO 100
1080 PRINT AT K,1;" "
1090 LET O=30
1100 RETURN
1200 CLEAR
1210 CLS
1220 PRINT AT 0,4;"KERSZ INFORMA
CIOT? (I/N)"
1230 LET D$=INKEY$
1240 IF D$="N" THEN GOTO 1247
1241 IF D$="I" THEN GOTO 1243
1242 GOTO 1230
1243 PRINT AT 10,0;"A JATEKOS CE
LJA VEGIGVEZETNI A BÁBUJÁT A VÁR
FALAN UGY, HOGY AZ NEUTKOZZON NEK
I A MOZGÓ AKADALYNAKES NE ESSEN
BELE AZ ÁRKOKBA SEM."
1244 PRINT
1245 PRINT "      LEPES AZ (L) B
ETUVEL.","      UGRAS AZ (M) BET
UVEL."
1246 PAUSE 4E4
1247 CLS
1250 RETURN
1290 REM
1300 REM
1310 REM
```



A középkori várfalon 10-szer kell végigvezetni a bábut úgy, hogy ne ütközzön a mozgó akadálynak, és ne essen bele az árokba se. Három élete van a bábunak; az irányítás módja benne van a programban.

```
  1 CLS
  2 GOSUB 2000
  10 LET M=3
  20 LET P=0
  21 LET Y=1
  30 LET L=1
  40 LET S=3
  45 LET B=1
  50 LET K=6
  60 LET S$="
  66 LET U$="
  65 PRINT AT 21,0;"
  70 LET A=11+INT (RND*5)
  75 PRINT AT 0,0;"BABU:";M;"
     MENET:1   PONT:00"
  80 PRINT AT S,0;S$;AT S,21;S$;
AT 11,0;S$;AT 11,21;S$;AT 19,0;S
$;AT 19,21;S$
  85 PRINT AT 4,0;U$;AT 6,0;U$;A
T 8,0;U$;AT 12,0;U$;AT 14,0;U$;A
T 16,0;U$;AT 20,0;U$
  90 PRINT AT K,L-1;"
  92 PRINT AT S,A;"
  94 IF L<9 OR L>19 THEN GOTO 10
0
  95 IF L=A OR L=A+1 OR L=A+2 TH
EN GOTO 99
  96 GOTO 1600
  99 LET L=L+B
 100 IF L=27 THEN GOTO 1500
 101 LET A=A+B
 102 IF A=10 THEN LET B=1
 103 IF A=16 THEN LET B=-1
 105 IF INKEY$="L" THEN LET L=L+
1
 120 IF INKEY$="M" THEN GOTO 100
0
 130 GOTO 90
1000 PRINT AT K,L;"  "
1010 LET L=L+1
1015 LET K=K-1
1020 PRINT AT K,L;"
1025 PRINT AT K,L+1;" "
1030 LET L=L+2
1040 LET K=K+1
1045 PRINT AT K,L;"
1050 GOTO 89
1500 PRINT AT S,A;"
1510 PRINT AT K,L+1;" "
1520 LET P=P+10
1525 PRINT AT 0,29;P
1526 IF P=30 OR P=60 THEN GOSUB
1900
1527 IF P=90 THEN GOTO 1580
1530 LET L=5+0
1540 LET L=1
1550 LET K=K+B
1560 LET A=11+INT (RND*5)
1570 GOTO 90
1580 PRINT AT 10,5;"MEGNYERTED A
 JATEKOT."
1581 PRINT AT 15,13;"UJAT?";AT 1
6,14 "I/N"
1590 IF INKEY$="I" THEN RUN
1591 IF INKEY$="N" THEN STOP
1592 GOTO 1590
1600 PRINT AT K,L+1;" "
1610 LET K=K+1
1620 IF K=20 THEN GOTO 1700
1630 PRINT AT K,L+1;"
1640 GOTO 1600
1700 FOR I=0 TO 9
1701 PRINT AT K,L+1;"
1702 FOR R=0 TO 2
1703 NEXT R
1704 PRINT AT K,L+1;"
1705 FOR R=0 TO 2
1706 NEXT R
1707 NEXT I
1708 PRINT AT K,L;"  "
1710 LET M=M-1
1712 IF M=0 THEN GOTO 1730
1713 PRINT AT S,A;"  "
```

```
1720 GOTO 20
1730 PRINT AT 10,4;"ELVESZTETTED
 A JATEKOT."
1731 GOTO 1551
1900 LET L=1
1910 LET K=-6
1911 LET S=-6
1912 LET Y=Y+1
1913 PRINT AT 0,18;Y
1920 RETURN
2000 PRINT AT 0,3;"KERSZ INFORMA
CIOT (I/N) ?"
2010 IF INKEY$="I" THEN GOTO 204
0
2020 IF INKEY$="N" THEN GOTO 220
0
2030 GOTO 2010
2040 CLS
2050 PRINT AT 5,0;" A JATEKOSNAK
 BABUJAVAL A FELSO"
2060 PRINT "SZINTROL KELL 3-SZOR
LEJUTNIA A"
2070 PRINT "LEGALSO SZINTRE. UTJ
A SORAN MIND"
2080 PRINT "3 SZINTEN EGY VIZSZI
NTESEN MOZOG"
2090 PRINT "LIFTEN KELL ATUTAZNI
A. HA EZ SI-"
2100 PRINT "KERUL, AKKOR GYOZ. H
A NEM, AKKOR"
2110 PRINT "VESZIT."
2120 PRINT AT 15,6;"LEPES AZ (L)
BETUVEL";AT 16,6;"UGRAS AZ (M)
BETUVEL"
2130 PAUSE 4E4
2135 CLS
2140 RETURN
2200 CLS
2210 RETURN
2300 REM
2310 REM
2311 REM IRTA KOVACS KAROLY VAC
```

PROGRAM-(ÖN)KRITIKA

Az 1984/6. szám AMÖBA programjának helyesbítése éppen a hiba. Erre a javításra semmi szükség nem volt. Aki próbálta, tapasztalta, hogy a program BASIC-ben rendkívül hosszú a sok művelet miatt.

Kezdőként írtam e programot, így a szervezése nem a legpraktikusabb. Gépi kódban csinálgatom a tökéletesített változatot, amely már elvileg működik.

A legbanálisabb hiba a programban az, hogy csak üres helyen ellenőrzi, hogy mellette elegendő mennyiségű jel van-e. És akkor, amikor az elkészülő „ötös" be van kerítve, nem reagál rá.

A javítások:

a 1320-as sor az eredeti marad

.
.
.

460 IF A%(I,J)=V% THEN GOSUB 1450

948 V%=215 : IF A%(I,J)=V% THEN GOSUB 1450

.
.
.

1450 X=1 : X1=1 : Y=0 : Y1=0 : GOSUB 1550

1460 X=-1 : X1=-1 : GOSUB 1550

1470 X=0 : X1=0 : Y=1 : Y1=1 : GOSUB 1550

1480 Y=-1 : Y1=-1 : GOSUB 1550

1490 X=1 : X1=1 : GOSUB 1550

1500 X=-1 : X1=-1 : Y=1 : Y1=1 : GOSUB 1550

1510 X=1 : X1=1 : GOSUB 1550

1520 X=-1 : X1=-1 : Y=-1 : Y1=-1 : GOSUB 1550

1530 RETURN

1550 F=0

1555 IF A%(I+Y,J+X)=V% THEN X=X+X1 : Y=Y+Y1 : F=F+1 : GOTO 1550

1560 IF F<4 THEN RETURN

1570 IF V% = 215 GOTO 956

1580 GOTO 530

törölni: 959; 960

Remélem, most már használható lesz.

STOCK PÉTER

Az olvasó írja

Egyre több, szakmai kérdéseket, véleményeket tartalmazó levelet kapunk, amelyek nagy része már nem nekem szól, hanem más rovatok szerkesztőinek. Ezért úgy tartom igazságosnak, hogy a rovat egy részében a levelekre a szerkesztőség munkatársai válaszoljanak. Ez a mostani tehát az utolsó hagyományos levelezési rovat, a következőkben már többen válaszolunk. Kérem olvasóinkat, fogadják a megújult rovatot a régi szeretettel.

Csordás János, Debrecen,

Kartács u. 6. 4032

Melyik Ön szerint a jobb gép egy amatőr számára, a Commodore 64 vagy a 48 k-s Sinclair Spectrum?

Fogas kérdés, nem is lehet rá egyértelműen válaszolni. A két gép között – ha valamelyik szomszéd országban vásárolja – árban nincs nagy különbség. Ha én gépet vennék, akkor a következő szempontok szerint választanék: van-e itthon megbízható szerviz, hozzá lehet-e jutni megfelelő programcsomagokhoz, milyen gazdag a perifériakészlet, lehet-e hozzá háttértárolót csatlakoztatni – főleg hajlékonylemezt –, sok ilyen gép van-e az országban vagy kevés stb. Ha diák lennék, az is érdekelne, hogy az iskolában van-e valamelyikhez hasonló gép; ha pénzt szeretnék keresni a géppel, akkor erősen befolyásolna, hogy az adott gépre van-e fizetőképes szoftverkereslet, tehát melyik gépet használják inkább a vállalatok, intézmények. Miután lapunk egyik cégnek sem ügynöke, a kérdésre végül is nem adok választ.

Botka Péter, Kiskunfélegyháza,

Irányi u. 2. 6100

Ha lehet, küldjenek (természetesen pénzért) egy számítógép-katalógust, amely árakat is tartalmaz, lehetőleg nyugatnémet márkában. Eddig csak HT–1080Z típusú gépen dolgoztam – ez van az iskolában –, így nincs semmilyen dokumentációm más gépekről.

Kérése kapcsán sokaknak elmondhatom, hogy csak a lapban tudunk ilyen kéréseket teljesíteni – néha. Akkor is a viszonylag hosszú átfutási idő miatt mire az árlista megjelenik, az árak már nem érvényesek (lásd az 1985/1. számban a linzi árakról írt cikkemet; mire megjelent, minden ár 10–30%-kal kevesebb volt). Nem tudunk alkatrész-beszerzésre vállalkozni, sem ismeretlen japán, hongkongi és más gépekhez magyar nyelvű leírást szerezni.

Kajdi János, Pécel,

Petőfi Sándor u. 36/a. 2119

Miután az utóbbi időben nagy gépen dolgozom, saját célú programjaimat csak papírprocesszoron futtathatom. Ezért arra az elhatározásra jutottam, hogy megpróbálom ismereteimet cikkleírással hasznosítani, ha erre igény van. Kérem, tájékoztassanak az ide vonatkozó játékszabályokról: a) mik a használható cikktémák, b) az igénybe vehető terjedelem, c) a benyújtás módja, d) módosítási jogok, ügyrend, szerzői jogok, e) az átfutási idő, f) a díjazás módja, mértéke stb., g) elfogadnak-e másodközlésre más lapban már megjelent cikket, változtatásokkal vagy anélkül, h) minden egyéb, ami ilyenkor még felmerülhet.

Válaszom valamennyi olvasónknak szól, várjuk írásaikat.

a) Lehetőleg gyakorlati kérdésekről írjanak, programokat küldjenek, szívesen veszünk programozástechnikai fogásokat, ötletes megoldásokat. Szeretjük közölni a számítógép-építők eredményeit, írásokat várunk például különféle amatőr perifériákról, egyszóval mindarról, amit olvasóink hasznosítani tudnak. Majdnem elfeledkeztem a tanárokról, az iskola – számítógép rovatban. Rájuk is számítunk.

b) Szeretjük a rövid, 5–8 gépelt oldalas írásokat, ábrákkal együtt.

c) A felelős szerkesztőhöz vagy valamelyik rovatszerkesztőhöz kell eljuttatni levélben vagy személyesen.

d) Kéziratot nem küldünk vissza. Ha módosítani kell, akkor azt a szerző hozzájárulásával tesszük. A kézirat beküldésével a szerzőtől jogot kapunk a megjelentetésre.

e) 3–6 hónap.

f) A megjelent cikkekért tartalomtól függően és a terjedelem szerint az előírásoknak megfelelően fizetünk.

g) Általában nem.

h) Nem szeretjük az idegen kifejezéseket. Ha nincs magyar megfelelő, akkor az idegen szavakat általában magyar fonetikával írjuk, illetve, ha van, akkor a megfelelő magyar kifejezést írjuk be, a szerző megkérdezése nélkül.

Csige József, Balmazújváros,

Kastélykert u. 11.

Nem tudom, mi lehet az oka, hogy rendszertelenül jelenik meg a lap, és az 1984/5. számot is csak 1985. februárban tudtam ügyvel-bajjal megszerezni. Végre megtaláltam benne az előfizetői levelezőlapot. Még aznap elküldtem, de csekket vagy választ eddig nem kaptam. Hónapokig tartó kereséssel sem tudtam megvenni az 1984/4. számot, pedig bejártam érte a környező településeket. Ha lehet, kérem, küldjenek részemre egy példányt belőle.

Az előfizetésre érkezett levelezőlapokat átadtuk a Postának. Arra kérem, reklamáljon, ti. – mindenkinek mondom – a következő évben előfizetőinknek különféle kedvezményeket fogunk adni, így nem mindegy, hogy a lapot példányonként veszi, vagy előfizetőként kapja. A 4. számból még van néhány, elküldjük.

Gócza László, Hatvan,

Dembinszky út 43/b. 3000

Az év elején láttam meg az újságárusoknál a µM 1985/1. számát. Sajnos a régebbi számokat azóta is hiába keresem. Szeretném, ha megírnák, hogy eddig hány szám jelent meg, és ha közzétennék a kérésemet, hogy a régebbi számokat – ha jó állapotban vannak, akár eredeti áron is – megvásárolnám.

Eddig (május közepén írom a válaszokat) összesen 9 szám jelent meg. Néhány régebbi szám még kapható a Neumann János Számítógéptudományi Társaságnál (Bp. V., Báthori u. 16.).

Szládovics Péter, Tata,

Lumumba út 11. 2890

Márciusban kaptam egy 1 k-s Sinclair ZX81-et. A gép nyelvezetét a HT–1080Z iskolaszámítógépen szerzett BASIC tudásommal pofon egyszerű volt megtanulnom. Egyre kínosabb viszont a BASIC lassúsága, mind a két gép esetében. Hosszabb játékprogramoknál még a HT is „csigának" bizonyult, nem is beszélve a ZX-ről. Ezúton szeretném megkérni a Szerkesztőséget és tisztelt olvasótársaimat, hogy ha lehet, küldjenek nekem a Z80 assembler használatáról pontos útmutatást, némi leírást a gépi utasítások jelentéséről és használatáról (elágazások, feltételes ugrás, feltétlen ugrás stb.), mivel ezen a téren még nem ismerem egyik gépet sem. Szívesen ismerkednék meg ZX81 és HT–1080Z tulajdonosokkal programcsere céljából. Bármilyen – játék-, oktató- esetleg adminisztrációs – program érdekel. Ilyeneket én is szívesen küldök.

Olvasótársak, tessék!

Asztalos László, Jihlava,

ul. Brtnická 13. Csehszlovákia CS–58601

Csehszlovákiában tanulok, így nincs lehetőségem arra, hogy egy otthoni számítógépes klubba beiratkozzam. A programozással végső célom, hogy számítógéppel vezérelt modellvasút terepasztalt készítsek. Ehhez még sokat kell tanulnom, de az „algoritmus" vázát már kigondoltam.

Kérte még az 1984/5. számot. Sajnos egyetlen darab sincs belőle. Ha már számítógéppel vezérli a vasutat,

akkor írja meg hogyan, mert több levelezőm kérte, hogy ilyen cikket is közöljünk.

Ódor József, Baja,

Eszperantó u. 3. 6500

Szeretnék KIT-számítógépet építeni. Ehhez szeretném megtudni, hogy hol lehet megrendelni vagy megvásárolni; mennyibe kerül; mikorra várható; mit tud várhatóan a gép, és hol, milyen formában lehet hozzájutni az építéséhez és üzemeltetéséhez szükséges dokumentációhoz, információhoz.

Amióta megírtam, hogy szeretnénk egy KIT-számítógépet létrehozni, nagyon sok levelet küldenek, sokan jelzik vételi szándékukat. El kell mondanom, hogy mindig újabb reménysugár jelenik meg, azután – elsősorban az alkatrész-beszerzés miatt – az ügy elbukik. Azután újabb remény, újabb bukás, de az erőfeszítést, hogy legyen, nem hagyjuk abba. Így a kérdéseire csak később fogok válaszolni.

Mocsi Péter, Bruty 119.

94355 Csehszlovákia

Sajnos az itteni lapok között még nincs az Önökéhez hasonló, amely ilyen széles körben tájékoztatná a számítástechnika iránt érdeklődőket. Az eddigi számokat is csak nagyon nehezen tudtuk megszerezni, és már szinte ronggyá olvastuk őket. A lap megrendelésével többször is próbálkoztunk a prágai Magyar Kultúránál, sajnos sikertelenül. Ezért szeretném megérdeklődni, milyen módon fizethetném elő a lapot.

Sajnos a külföldi terjesztés vontatottan megy, még nem sikerült az igazi felelőst megtalálnunk. Keressük!

Egyed Gábor és Lestyán Attila,

Kecskemét, pf. 320/f. 6004

Tudjuk, hogy léteznek µklubok, de azt nem, hogy hol. Szeretnénk megkapni a címüket. Ha lehet, kérjük küldjék el nekünk a számítógép-építő klubok címét is, talán tudnának nekünk segíteni személyi számítógép építésében.

A µklubok adatait a µM különkiadásaként rövidesen megjelentetjük. Szeretném, ha a még be nem jelentett közösségi vagy magán µklubok elküldenék adataikat (a klub neve, címe, az üzemeltető intézmény neve, a klub vezetőjének neve és telefonszáma, milyen gépekkel rendelkeznek, van-e tagsági díj és mennyi), hogy teljes címlistát tudjunk közölni.

Baranyai Attila, Győr,

Heszky Erzsébet u. 10. 9024

Második gimnazista vagyok. Iskolánkban 1984 őszén indult a számítástechnikai szakkör kezdőknek. Nekünk nagyon tetszenek a lapjukban lévő számítógépekkel kapcsolatos programozások, de mi még ilyen szinten nem tudunk programozni. Kérnénk Önöket, hogy közöljenek játékprogramokat Commodore 64-es gépekre is, mert az iskolában eddig csak 2 db játékprogramunk van. A C64-es gépen kívül 6 db VC–20-asunk is van, de csak 3–4 programot tudtunk hozzájuk szerezni.

Azt hiszem, nincs ok panaszra, elég sok programot hozunk Commodore gépekre.

Várszegi Attila, Budapest,

Leányka u. 3. 1221

Tanuló vagyok, VC–20-as gépem van. Néhány észrevételemet közlöm a µM-nal kapcsolatban. 1. Kevés a lapban a programleírás, és ha van, nehezen olvasható. 2. Hiányolom a Commodore 64 és VC–20-ra írt programokat, pedig ez nálunk eléggé elterjedt géptípus. Sokat foglalkoznak viszont kevéssé gyakori gépekkel, például az ABC–80-nal. 3. Jó lenne széles körben elterjeszteni a játék- és oktatóprogram-kazettákat.

1. A programleírásokat mátrixnyomtatón készítjük, sajnos csak ilyen minőséget tudunk előállítani. 2. Lásd Baranyai Attilának adott válaszomat. ABC–80-ra egyre kevesebb programot kapunk és közlünk. 3. Ezzel igen sok bolt (például a Skála Metró) foglalkozik.

Gilvázi István, St. Knicanina 48.

21220 Becej, Jugoszlávia

Becsén, egy kis Tisza menti városka számítóközpontjában dolgozom, mint programozó. Ezenkívül fő tevékenységem a mikroszámítógépes kultúra terjesztése környezetemben, aminek eredményeként megalakult a becsei BIT nevű mikroszámítógép klub. Főleg az iskolákba szeretnénk bevezetni a technika eme újdonságát, de persze ez nem megy problémamentesen.

Szeretném valami módon felvenni a kapcsolatot Önökkel, mint központi klubbal. A mi kis klubunknak jelenleg egy Spectrumja és 10 db Hobby–ZR–84-es gépe (jugoszláv gyártmány, egy maszek kisiparos terméke), valamint egy Nascom–2 típusú gépe van. A ZR–84-es teljesen kompatibilis az amerikai TRS–80-nal, illetve a magyar iskolaszámítógéppel is. A ZR–84-esek mindössze 1 hónapja léteznek, ezért nincs valami szörnyű nagy tapasztalatunk, de a külföldi könyvekből és másfajta géptípusokról átírt, valamint saját programtermékeinket összevetve, kb. 200 rövidebb-hosszabb programunk van rá.

Becse testvérvárosa Szekszárd, ezért szeretnénk az ottani gimnáziummal felvenni a kapcsolatot program- és tapasztalatcsere céljából. Ehhez kérjük a segítségüket.

Szívesen küldenénk programokat az újság részére is, amely végre előfizethető a FORUM könyvesboltjain keresztül.

Hosszú leveléből csak részleteket tudok közölni. Az együttműködési javaslat számunkra nagyon megtisztelő, létrehozzuk!

Németh Zoltán, Veszprém,

Komócsin u. 38/c. 8200

A Környezetvédelmi Intézetnél dolgozom, mint vizsgáló üzemmérnök. Munkám során M08X típusú professzionális személyi számítógéppel dolgozom, programokat alkalmazok és készítek is. Tudásomat – a BASIC nyelv ismeretét – autodidakta módon, önállóan sajátítottam el. Szeretném azonban számítástechnikai ismereteimet elmélyíteni a számítógépalkalmazás területén, de sajnos nagyon kevés információm van arról, hogy erre milyen lehetőségek vannak. Vagyis hogy milyen intézmények foglalkoznak felsőfokú levelező képzéssel e téren, és a gyakorlati alkalmazás szempontjából melyiket érdemes választani. Erre vonatkozó tanácsát előre is köszönöm.

A kérdésre nehéz válaszolni. Ajánlatokat adhatok, de többet kellene tudnom arról, hogy milyen területen kívánja tudását hasznosítani (hardver, szoftver, alkalmazás, netán szervezés), akkor talán pontosabban tudnék válaszolni. Azt javaslom, hogy kérjen a SZÁMALK tanfolyamokról leírást (Bp. XI., Szakasits Árpád út 68.), azután keresse meg a Marx Károly Közgazdaságtudományi Egyetemet (Bp. IX., Dimitrov tér 8.), a Budapesti Műszaki Egyetemet (Bp. XI., Műegyetem rkp. 3/9.), a Kandó Kálmán Műszaki Főiskolát (Bp. III., Nagyszombat u. 19.), a Pénzügyi és Számviteli Főiskolát (Bp. XIV., Buzogány u. 10/12.), de ma már szinte valamennyi főiskolán és egyetemen, hol jobban, hol kevésbé alaposan tanítják a számítástechnikát.

Kókai László, Biatorbágy,

Búzavirág u. 2. 2051

Valahol a tévében hallottam, hogy meg fog jelenni egy könyv, amiben csak programok lesznek. Tessék megírni, hogy kb. mennyibe fog kerülni, és hol lehet majd kapni.

Ahogy mondani szokták, a hír igaz, de a kérdésre még nem tudok válaszolni. Ez a könyv lesz a µKönyvtár első kötete, a kiadás fortélyait most tanuljuk. A következő, ha a kiadói jogokat megkapjuk, egy híres számítástechnika-történeti mű lesz, H. Goldstine: A számítógép Pascaltól von Neumannig.

●

A sok levél és a kéthavonkénti megjelenés miatt olyan leveleket is közöltem – mivel véleményem szerint tartalmuk sok olvasónkat érdekelheti –, amelyeket valamikor az év elején kaptunk. A késedelmes válaszért szíves elnézésüket kéri:

KOVÁCS GYŐZŐ

SZÁMORG

Számítástechnikai Kisszövetkezet

VÁLLALJUK

termelőszövetkezetek, állami gazdaságok, iparvállalatok, kereskedelmi vállalatok, városgazdálkodási vállalatok, tanácsok és egyéb gazdálkodó szervek számára

ügyvitelszervezési,
rendszerszervezési,
tervezési és programozási

feladatok mikroszámítógépekre (ROBOTRON, Sirius, Victor, IBM, Alpha, Micro, VT20/A, VT20/N, Commodore 64, 610, 720, stb.) való elvégzését.

Munkatársaikat betanítjuk!

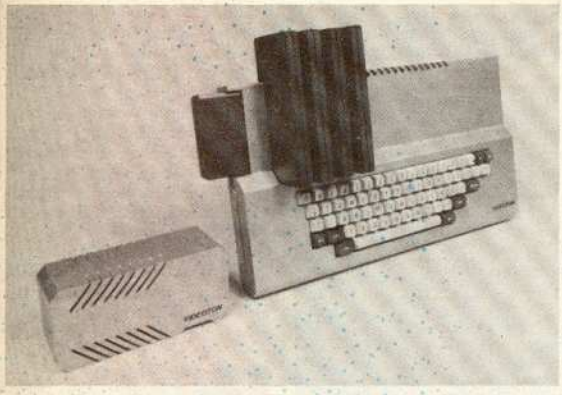
Rendelje meg nálunk
főkönyvi könyvelés, analitikus anyag-, fogyóeszköz-, termék-nyilvántartási, bér- és munkaügyi rendszerek, személyzeti nyilvántartás, állóeszköz-nyilvántartás, tervezés, statisztikai feldolgozások, egyéb célrendszerek szervezését, programozását!
A szervezések bevezetését felügyeljük, a programokra **egyéves garanciát** adunk.
A számítógépek beszerzésében is közreműködünk.

INGYENES SZAKTANÁCSADÁS!

Számítógépek illesztését megtervezzük, elvégezzük.
Személyes konzultációt biztosítunk.

Címünk: **Budapest VIII., Rákóczi út 19. I. emelet**
1368 Budapest Pf. 237. SZÁMORG

Telefon: **33-55-77, 340-706**
Telex: **22-5241**



Videoton újdonságok

Az év második felében két termékéből is jelentősebb mennyiséget kíván piacra hozni a Videoton. 2000 darabot szeretne forgalomba hozni TV Computeréből, a kifejezetten vállalati célokra tervezett VT16 gépből pedig 800 darabot.

A végleges konstrukciójában az idei BNV-n látott TV Computer több figyelemreméltó sajátossággal bír. A teljes magyar betűkészlet igen jó minőségű megjelenítése mellett lehetővé teszi nagyobb felbontású video monitor alkalmazását is. Az utóbbi esetben soronként 64 karaktert tud alfanumerikusan megjeleníteni (összesen 24 sor), illetve 240 × 512 pontos, kétszínű grafikát támogat. A géppel együtt bemutatták azt a színes tévékészüléket is, amely rendelkezik a megfelelő adapterrel és monitor üzemmódban használható. A gép átgondolt konstrukcióját jól mutatja az igen ügyesen kialakított, max. 4 kártyás bővítési lehetőség.

A gép billentyűzete akár a professzionális igényeknek is tökéletesen megfelel. Legmeglepőbb azonban az ára: 32 kbájt RAM kapacitású alapváltozata mindössze 25 ezer forintba kerül majd, hálózati tápegységgel együtt.

A VT 16 gép egykártyás kialakítású, két üzemmóddal bíró személyi számítógép. 8 bites üzemmódban a CP/M-nek megfelelő operációs rendszer alatt futtathatók a programok. 16 bites üzemmódban az MS–DOS-nak megfelelő operációs rendszer alatt működő programokat lehet használni. Ez jelenti az IBM személyi számítógépen működő programok egy részét is, de nem minden ilyen programot, mivel a VT16 konstrukciója némileg eltér az IBM PC-étől. A memória kapacitása 256 kbájt RAM, a beépített floppyk egyenként 1 Mbájtosak, és 10 Mbájtos Winchester lemezzel is szállítják a gépet. A konstrukciós előnyök mellett itt is az ár a leginkább figyelemreméltó. A 2 Mbájt floppy kapacitással bíró változat 360 eFt-ba kerül, míg a Winchester lemezes mindössze 590 eFt-ba.

Winchester

Bemutatták az első, szocialista országban készült Winchester-elvű tárolótípust, MW–1000 típusjelzéssel. A Magyar Optikai Művek Winchester-elvű tárolócsaládjának ez az első tagja. Mérete 5,25", tárolási kapacitása formattált esetben 10 Mbájt. Az adatok 6 tárcsaoldalon helyezkednek el, minden tárcsaoldalon 440 funkcionális és 10 parkolósávot tartalmaz. A mikroprocesszorral szabályozott meghajtómotor gyors felpörgést és nagy fordulatszám-stabilitást biztosít a kívánt 3600 fordulat/perc sebességnél. A külön felépített intelligens léptetés gyors és pontos sávkeresést tesz lehetővé, így az átlagos elérési idő 100 ms. A meghajtó külső méretei megegyeznek az 5,25" hajlékonylemezes tároló méreteivel.

Optikai billentyűzet

Új magyar szabadalom alapján optikai billentyűzetet mutattak be májusban, DCD–OT–327 néven. A 96 billentyűhelyből álló infrafény-letapogatásos eszköz teljesen elektronikus vezérléssel működik, mechanikai érintkezőket nem tartalmaz. Megbízhatósága igen nagy, élettartama hosszú. Tetszőlegesen programozható karaktergenerátora révén a kívánt kód a felhasználó igénye szerint választható. A billentyűzet antisztatikus, ütésálló műanyagból készült, formatervezett háza az asztallap síkjához viszonyítva dönthető, az alul elhelyezett, bepattintható tartóláb segítségével.

Az optikai elvű billentyűzet további előnye, hogy kb. felébe kerül a hagyományos billentyűzeteknek. Ez indokolja, hogy ebbe már Primo gépet is szereltek, melyet Primodata néven mutattak be.

A legolcsóbb

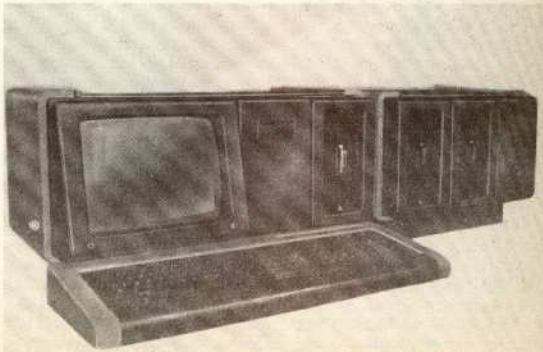
Tavasszal mutatták be a szocialista országok legolcsóbb mátrixnyomtatóját, a DCD–PRT–42-t, melynek ára 24 ezer forint. A mindössze 3,4 kg-os nyomtató 9 × 9 pontból álló mátrixszal képezi a nyomtatandó jeleket. Az egy sorba nyomtatott karakterek száma a betűszélességtől függően 42 vagy 80, a nyomtató sebessége is ettől függően változó, sorrendben 50 vagy 100 karakter/perc. A befűzhető papír maximális szélessége 210 mm.

A mátrixnyomtatónak grafikus üzemmódja is van. Elektronikus vezérlésű, érzékeli a papírhiányt. Elsősorban a Primo mikrogépekhez fejlesztették ki, azért is nyomtat 42 karaktert egy sorba, mert a Primo is ennyit jelentet meg a képernyő egy sorában. Az optimálisan választható gazdag illesztőkészlet, például a Centronics vagy a V.24 azonban lehetővé teszi bármely hazai mikroszámítógéphez való illesztését is.

Az A 5120 – 16 bitesre bővítve

Hazánkban is jól ismertek az NDK gyártmányú A 5100-as professzionális mikroszámítógépek. Ezek közül is az 5120 típus a leginkább közkedvelt. 1985-ben az eredeti formatervezésű képernyős megjelenítőbe építve egy új, nagy teljesítményű változata jelent meg. Az eddigi Z80 analógiájú nyolcbites mikroprocesszor mellett egy 16 bites, az NDK-ban gyártott mikroprocesszort is beleépítettek, amely kompatibilis a Z8000-rel. E nagy teljesítményű mikroprocesszorhoz nagy kapacitású, 256 kbájtos operatív tárat is biztosítanak. Viszont ezek az erőforrások teljesítményben messze túlszárnyalják a géphez kapcsolt háttértárakat, mivel mindössze 3 × 0,5 Mbájt közvetlen hozzáférésű kapacitást biztosító hajlékonylemezes tároló csatlakoztatható.

A gép nyolcbites része továbbra is a CP/M-szerű SIOS operációs rendszer alatt fut; a 16 bitesnek az operációs rendszerét MUTOS–U-nak nevezik, és Unix-kompatibilis.



LÉPÉSRŐL LÉPÉSRE

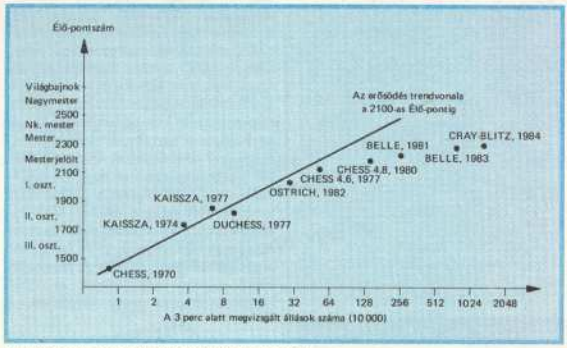
Az előzőekben bemutattuk azokat az alapgondolatokat, elveket, amelyekre a jelenlegi legerősebb sakkprogramok épülnek. Ezek az elvek nagyon kevéssé voltak „emberszabásúak"; nemigen hasonlítottak arra, ahogy az ember sakkozik.

A sakkprogramok eddigi fejlődésének trendje

Elvileg semmi sem zárja ki, hogy az ember gondolkodásmódjától alapvetően különböző elvek alapján az embernél lényegesen erősebben sakkozó számítógépet lehessen készíteni. Például járművek építésére az ember olyan elveket és eszközöket (kereket vagy sugárhajtást) használ, amelyek a természetben egyáltalán nem fordulnak elő, és a kívánt működést ilyen velejéig mesterséges módon lényegesen eredményesebben tudja megvalósítani, mint ha mondjuk bonyolult ízület- és csuklórendszerekből építene galoppozó vagy ügető autót.

A sakkprogramok játékerejének kezdeti látványos felfutása azt sejtette, hogy ezen a területen is hasonló lesz a helyzet: a számítógép speciális lehetőségeinek ügyes kihasználásával, szellemes algoritmusok segítségével sikerülni fog világbajnok erejű vagy még erősebb programot készíteni. Sorra születtek a jóslatok, hogy mikor fogja a számítógép legyőzni a világbajnokot. Az általam ismert, komoly szakértőktől származó legkorábbi jóslat 1959-ből való, és 1965-öt jelölte meg, mint a világbajnok erejű program megjelenésének időpontját. Jóslatokban azóta sincs hiány, de a legerősebb sakkprogram sem közelíti még meg a nemzetközi mesterek, és még kevésbé a nagymesterek szintjét.

A jóslatok mégsem voltak teljesen megalapozatlanok. Az *ábrán* bemutatjuk a nagyszámítógépekre írt sakkprogramok erősödésének tendenciáját a használt gépek működési sebességének függvényében. A programok erejét a szokásos nemzetközi Élő-pontszámmal mérjük a függőleges tengelyen, a használt gépek sebességét pedig azzal, hogy három perc alatt hány állást képes megvizsgálni. Átlagosan ennyi idő jut ugyanis egy-egy lépésre egy versenypartiban. A sebességet a vízszintes tengelyen jelöljük, logaritmikus ábrázolásban, tehát a vízszintes tengelyen minden osztás az előzőhöz képest dupla sebességet jelent.



A sakkprogramok erősödésének trendje

A programok erősödésének trendje kb. 1977-ig – vagy fogalmazzunk inkább így: kb. a 2100-as Élő-pontszámig – eléggé egyenletes. Ennek alapján joggal gondolhatták a szakértők, hogy lényegében a számítógépek gyorsulásának trendje alapján jó jóslatot tehetnek a sakkprogramok erősödésére. Ez a trend 1985-re valóban elérte volna a világbajnok szintjét.

Az ábrán azonban az is jól látható, hogy a 2100-as Élő-pontszámnak megfelelő játékerő közelében az addigi egyenletes trendvonal élesen megtörik: az utóbbi hét évben a sakkprogramok erősödése nem tartott lépést a számítógépek gyorsulásával. (Jegyezzük meg, hogy a számítógépek gyorsulásának trendje egyelőre még nem tört meg.)

Ez a mérési mód természetesen eléggé pontatlan, hiszen, mint a korábbiakból tudjuk, az adott idő alatt megvizsgálható állások száma nemcsak a számítógép sebességétől függ, hanem több más tényezőtől, így a kiértékelő függvény helyes felállításától, a programozási fogásoktól is. Hogy mennyire így van, mutatja a mikrogépeknek a legutóbbi években tapasztalható látványos fejlődése: noha sebességük a leggyorsabb óriásgépekének legfeljebb a századrésze (3 percenként 200 ezer), a legjobbak játékereje szintén megközelíti a 2100 Élő-pontot, és ha nem is versenyképesek Belle-lel vagy Cray-Blitz-cel, de erősebbek számos, náluk tízszer vagy ötvenszer gyorsabb, óriásgépre írt programnál.

Mégis, a trendvonal ilyen látványos megtörése elgondolkodtató. A 2100-as Élő-pontszám már jelentős színvonalat képvisel, megfelel a magyar mesterjelölti minősítésnek. A mesteri vagy még inkább a nemzetközi mesteri szintet áttörni azonban az emberek közül is csak keveseknek sikerült.

Úgy tűnik, a minimax algoritmus – sokféle továbbfejlesztésével és kiegészítésével együtt – egyfelől nagyon hatékonynak bizonyult, hiszen emberi mércével mért sakktudás nélkül is lehetővé tette egy elég magas szint elérését, másfelől azonban távolról sem bizonyult annyira hatékony mesterséges eszköznek, mint a kerék vagy a sugárhajtás, hiszen az ember csúcsteljesítményeit eddig nem sikerült megközelíteni vele, és amint az ábra mutatja, kétséges, hogy valamikor is fog-e sikerülni.

Ha ez így van, két út kínálkozik: vagy ki kell találni egy, a minimax algoritmustól radikálisan különböző játékalgoritmust, függetlenül attól, hogy az ember milyen „algoritmusokat" használ sakkozás közben, vagy jobban meg kell ismerni az ember gondolkodásmódját, és megpróbálni annak szellemében készíteni algoritmusokat. Az előbbi irányban tudomásom szerint eddig még nem született átütő erejű új ötlet, az utóbbi irányban azonban ígéretes eredmények mutatkoznak, amint azt a mikrogépek említett fejlődése is bizonyítja.

Mekkora fát vizsgál az ember?

Meglepően kevés különbség van egy nagymester és egy amatőr játékos között, ha csak olyan mutatókat vizsgálunk, hogy milyen mélyen kombinálnak előre, vagy hogy hány variációt vizsgálnak meg. Ezt a nagymesterek mindig is állították, és *de Groot* holland pszichológus és sakkmester már a harmincas évek végén tudományos pontossággal kimérte. De Groot nagymester-alanyai a kor legjobb sakkozói, az 1938-as híres AVRO-verseny résztvevői voltak, helyezésük sorrendjében: Keresz, Fine, Botvinnik, Aljechin (az akkori világbajnok), Euwe, Reshevsky, Capablanca és Flohr.

Egy tipikus vizsgálati eredményt mutat a *táblázat*. De Groot bonyolult állásokat vizsgáltatott alanyaival, hogy válasszák ki a szerintük legjobb lépést. A választott lépés erősségét 1-től 9-ig értékelte. Ez az egyetlen olyan sor a táblázatban, ahol igazán lényeges különbség fedezhető fel a nagymesterek és az amatőrök között: a nagymesterek általában megtalálták a legjobb lépést, az amatőrök távolról sem.

De Groot kísérletének eredménye

	Nagymesterek átlaga	I. oszt. amatőrök átlaga
Döntési idő (perc)	9,6	12,8
A megvizsgált első lépések száma	4,2	3,4
Max. vizsgált mélység (féllépés)	6,8	6,6
Összes figyelembe vett lépések száma (különböző mélységekben)	35,0	30,8
A választott lépés értéke	8,6	5,2

Könyvek

Az emberek által megvizsgált, átlagosan harminc-egynéhány pontból álló fa sok nagyságrenddel kisebb, mint amekkora fákat a sakkprogramok képesek elemezni, és az a néhány tízmilliónyi pontból álló fa, amit a programok képesek elemezni, mint korábban láttuk, csak elenyésző része a sakkjáték teljes fájának.

A különbség az emberi sakkozó és ezen belül is a nagymester javára egészen más pontokon, nem mennyiségi, hanem minőségi jellemzőkben jelentkezik. Például már mindjárt az állás észlelésénél: ha egy nagymester ránéz egy állásra, egész mást lát, mint az amatőr. Ennek igazolására bemutatunk egy érdekes kísérletet, aminek a tanulságai igen messzire vezetnek, és a sakkprogramozás számára is irányt mutathatnak.

A kísérlet lényege, hogy a kísérleti személyek néhány másodpercig nézhetnek egy sakkállást, majd a tábláról lesöpört állást kell rekonstruálniuk. A nagymesterek ezt a feladatot sokkal jobban tudták végrehajtani, mint az amatőrök, de ez még nem meglepő. Érdekesebb, hogy lényegesen megnőtt a nagymesterek fölénye, ha nemcsak a rekonstruálás során elkövetett hibák számát vették figyelembe, hanem azt is, hogy a hiba mennyire változtatta meg az állás jellegét. (Például egy „érdektelen" gyalogot egy mezővel odébb helyezni kisebb hiba, mint egy futót, amely ettől „jó" futóból „rossz" futó lesz.)

Még érdekesebb eredményt adott egy szellemes kontrollkísérlet. Véletlenszerű, dobókocka segítségével létrehozott állásokat rekonstruáltattak a kísérleti alanyokkal. A véletlen állások éppen csak megfeleltek a sakk szabályainak (arra azért vigyáztak), de értelmük nem volt. Ebben a kísérletben a nagymesterek és az amatőrök teljesítménye között semmi különbség nem mutatkozott. Sőt, a helyesen rekonstruált figurák darabszáma kevesebb volt, mint az értelmes (játszmákból vett) állásokban a kezdő sakkozók által helyesen rekonstruált figurák száma.

Ebből látszik, hogy az ember már első ránézésre sem a figurák konkrét helyét érzékeli, hanem a köztük fennálló kapcsolatokat, stratégiai értelemben értelmes relációkat. A nagymester attól nagymester, hogy számtalan ilyen relációt ismer. Hogy mégis körülbelül mennyit és miféléket, arról a következő folytatásban szólunk.

MÉRŐ LÁSZLÓ

(Folytatjuk)

Dr. Gánti Tibor: Chemoton elmélet (A fluid automaták elméleti alapjai)

(Budapest, 1984. OMIKK, 222 oldal. Ára: 270,– Ft.)

A chemoton elmélet az élő rendszerek mibenlétének elméleteként indult, majd lassan a fluid (kémiai) automaták általános elméletévé szélesedett. Nem elvont matematikai elmélet, hanem mérnöki, azaz azt vizsgálja, hogyan lehet fluid automatákat kémiai reakciórendszerekből konstruálni és az ilyen automaták hogyan működnek. Ezért nemcsak a konstrukcionális alapelveket tárgyalja, hanem konkrét példákat, sok esetben pedig iparilag megvalósított technológiai példákat is mutat be az egyes konstrukciókra.

E monográfia első kötete megismertet a fluid automaták „alkatrészeivel" és azok kapcsolási módjaival: az önreprodukáló alkatrészekkel és a belőlük felépíthető önreprodukáló, majd szaporodó fluid masinákkal, köztük a legérdekesebbel, a szaporodó programvezérelt fluid automatával, a chemotonnal. A könyv utolsó fejezete bemutatja, hogyan lehet a chemotonokból speciális számítógép-elemeket kifejleszteni, és felvázolja egy működési alapelveiben is teljesen új, „gondolkodó" computer, a kogitátor konstrukcionális elveit.

Az élő rendszerek működésének, keletkezésének és az élővilág kifejlődésének a magyarázatával a monográfia második kötete foglalkozik a chemoton elmélet alapján.

Szabó Szabolcs: Pszichologika

(Budapest, 1984. SZÁMALK, 2. kiadás, 144 oldal. Ára: 40,– Ft)

A könyv arról szól, hogy

1. – CSAK SENKIBŐL LEHET VALAKI.
   – AKI MÁR VALAKI, AZ VEZETHET.
   – AKI VEZET, NEM MINDIG ISMERI A CÉLT.
   – AKI ELJUT VALAHOVÁ, NEM BIZTOS, HOGY TUDJA, HOL VAN.
   – AKI TUDJA, HOL VAN, AZ MÁSHOL SZERETNE LENNI.
   – AKI MÁSHOL VAN, AZ TANÁCSOT AD, HOGY HOVÁ MENJENEK.
   – AKI ELFOGADJA A TANÁCSOT, AZ TANÁCSTALAN, EZÉRT FŐNÖKKÉ SZERETNE VÁLNI.
   – AKI FŐNÖKKÉ LESZ, AZ HATALOMRA ÁHÍT.
   – AKINEK HATALMA VAN, ANNAK NINCS ESZE, MERT MINDENE NEM LEHET.
   – AKINEK NINCS ESZE, AZ SENKI.

GO TO 1.

(A Dobó Andor és Kertész Ádám által alkotott ciklus a Pszichologika című könyv hátsó borítójára kívánkozik. Ezért ajánljuk a kedves olvasóknak, hogy saját példányukra kézirással vessék is föl a fentieket, így a könyv külseje végre megkapja a tervezői által megálmodott, belsejéhez illő formát. Ezáltal a mondanivalója kivetül, de remélhetőleg SENKI-nek sem szúr szemet.)

Dr. Dobay Péter – Dr. Poór József: Irodai szövegfeldolgozási rendszerek automatizálása

(Budapest, 1985. SZÁMALK, 164 oldal. Ára: 60,– Ft.)

A szövegfeldolgozásról – napjaink számítástechnikai és szervezési szakirodalmának egyik „slágeréről" – magyar nyelven eddig még nem jelent meg könyv. A szerzők úgy dolgozták fel a témát, hogy a leírtak a lehető legkönnyebben alkalmazhatók legyenek a mindennapi gyakorlatban. Ennek érdekében úgy mutatják be a szövegfeldolgozás alapfogalmait, eljárásait, módszereit, tevékenységeit és az automatikus szövegfeldolgozás tipikus rendszereit, hogy a hangsúlyt a szervezési, szervezeti, üzemeltetési szempontokra, azaz a megvalósításra helyezik.

Érénye a könyvnek, hogy külön fejezetet szentel a hazai lehetőségeknek: ismerteti a hazai piacon kapható szövegautomatákat, mikrogépeket, szövegfeldolgozási programcsomagokat, és ezek elemzésén alapuló javaslatokat ad a továbblépésre, a fejlődési irányokra.

Lipovszki György–Subai László–Beszeda Tamás: FORTH programozási rendszer és nyelv

(Budapest, 1985. LSI Alkalmazástechnikai Tanácsadó Szolgálat, 245 oldal. Ára: 175,– Ft.)

A szerzők elsősorban azoknak írták könyvüket, akik a FORTH programozási nyelvvel magyarul szeretnének megismerkedni. Néhány angolul megjelent könyv után ez az első, oktató céllal írt, magyar nyelvű FORTH könyv.

A bevezetésben a szerzők ismertetik a FORTH programozási nyelv és rendszer történetét, majd megismertetnek annak alapfogalmaival és az alapvető definíciókkal, a vezérlési struktúrákkal, az adatbehozatali és -kiviteli műveletekkel, a speciális utasításokkal, a FORTH szövegszerkesztővel és assemblerrel, végül bemutatják a FORTH használatát néhány, Magyarországon is ismert számítógépen.

A téma iránt részletesebben is érdeklődő olvasók irodalomjegyzéket is találnak a könyv végén a legfontosabb FORTH publikációkról.

Marschik Iván: Mikrogéprendszerek fejlesztése I-II.

(Budapest, 1985. SZÁMALK, kb. 800 oldal. Ára: 439,– Ft.)

A könyv jelentős hiányt pótol több területen is. Hasznosan forgathatják a személyi számítógépek tulajdonosai, hiszen segítségével jobban megismerhetik gépüket, és jobb, hatékonyabb programokat tudnak írni. Hasznos a szakemberek számára is, mivel több más tudnivaló mellett jelenleg csak itt található jó magyar nyelvű leírás az Intel 16 bites elemcsaládjáról, a CP/M és MP/M operációs rendszerről, a mikrogépes fejlesztőrendszerekről és a konkurrens programozásról.

A hazai szakirodalomban elsőként vezeti be és használja következetesen az absztrakt gép koncepcióját, amely jelenleg a korszerű rendszerfejlesztések alapja. Ennek segítségével a felhasználói mikrogéprendszerek, a fejlesztőrendszerek és a rendszerprogramok egységes áttekintése nem okoz nehézséget.

Az első kötet (Rendszertechnika) 6 fejezetből áll. A legnépszerűbb mikroprocesszorok tárgyalása után a memóriák és beviteli/kiviteli elemek ismertetése következik. A szerző külön fejezetet szentelt a sinrendszereknek, hangsúlyozva ezzel a mikrogéprendszerek kommunikációjának egyre növekvő fontosságát. Az utolsó fejezetben egy korszerű, működő mikroszámítógép-rendszer ismertetése és a többprocesszoros rendszerek jellemzése található.

A második kötet (Programrendszerek) 3 fejezetre tagolódik. Először a fejlesztőrendszerekről olvashatunk. Itt nagy súllyal szerepel a szimuláció és emuláció tárgykörre, ami a gyakorlatban egyre nélkülözhetetlenebb. A monitorokkal és operációs rendszerekkel foglalkozó fejezet egyrészt egy szinte teljes monitor részletes leírását tartalmazza, másrészt a CP/M és MP/M operációs rendszerek ismertetését, valamint jó néhány részletet a konkurrens rendszerekről. Az utolsó fejezet a fordítóprogramokkal és programnyelvekkel foglalkozik, igen sok példával.

A könyv használát közel 300 kép és ábra segíti. Meg kell említeni az igen hasznos és fontos magyar–angol és angol–magyar szakmai kisszótárt, a jó és részletes tárgymutatót, továbbá a szerencsésen megválasztott függelékeket.

A korszerű, hasznos információkat tartalmazó könyv stílusa könnyen érthető és követhető; a tanuló szintről határozottan, következetesen vezet el a jelentős szakmai felkészültséget igénylő problémákig.

HERNECZKI ISTVÁN

MA BIZTOSÍTJUK A HOLNAP TECHNIKÁJÁT

MŰSZERTECHNIKA KISSZÖVETKEZET

az alábbi mikroszámítógép típusait ajánlja az érdeklődők figyelmébe

MULTI CENTER

- 8 munkahely
- 256 K RAM
- háttértár Winchester

MXT

- IBM XT kompatibilis számítógép

MXT–M

- IBM XT kompatibilis Multifeladatos számítógép

MULTI WS

- VME bus kompatibilis több processzoros több munkahelyes CAD/CAM rendszer

TRANSMIC–8

- hordozható számítógép

TZ–80

- asztali számítógép

– Standard operációs rendszerek és programcsomagok használata.
– Vállaljuk kulcsrakész rendszerek szállítását egyedi igények szerint. Számítógépeinket, szoftverjeinket megtekinthetik, megvásárolhatják, lízingelhetik COMPUTER BOLT-unkban.

1076 Budapest, Majakovszkij u. 1/D. TEL.: 423-423
Nyitva: H–P. 9–17 óráig

SZÁMÍTÓGÉPES
KIÍRÓBERENDEZÉSÉHEZ

**LEPORELLÓRA
VAN SZÜKSÉGE?**

IGÉNYÉVEL FORDULJON
SZAKÜZLETÜNKHÖZ!

**A PÁTRIA NYOMDA
NYOMTATVÁNYELLÁTÓ KIRENDELTSÉGE
VÁRJA VÁSÁRLÓIT!**

SZÁMÍTÁSTECHNIKAI SZAKÜZLETÜNK AJÁNLATA:

- számítógépek sornyomtatóihoz leporellók széles választékban
- számítógépes pénzforgalmi nyomtatványok (átutalási és beszedési megbízások)
- komputer etiket címkék
- peremlyukkártyák, lyukszalagok (5 és 8 csatornás)
- telextekercsek (1, 2 és 3 példányos)
- pénztár- és összeadógépszalagok
- papír — írószerek

Cím: Budapest XIII.,
Csanády u. 5.

Nyitva tartás:
Hétfő—Péntek: 7.30—15.30-ig
Szombat: zárva

Telefon: 298-089

VÁLASZOK A TUDÁSPRÓBA KÉRDÉSEIRE

1. Nem kizárólagosan. Az alfanumerikus jelzőt a gyakorlat nem szó szerinti értelemben használja. A betűkön és a számokon kívül például minden más jel, ami szerepel az írógépek billentyűzetén, szerepelhet alfanumerikus jelekben is. Ma ennek a szónak a jelentésében a „nemcsak numerikus és nemcsak alfabetikus" tartalom dominál. E fogalom nem tekinthető egzaktnak, inkább kereskedelmi, mint tudományos fogalom. Az alfabetikus és numerikus jel fogalma sem teljesen tisztázott. Alfabetikus jel-e például az egyes szövegekben nélkülözhetetlen felső vessző (az ún. aposztrof)? A szakszótárak és lexikonok nem adnak egyértelmű és egységes eligazítást, arra vonatkozóan sem, hogy numerikus jel-e például a számok egészrészét a törtrésztől elválasztó, számírásnál nélkülözhetetlen pont vagy vessző, a hatványozás felfelé mutató nyila.

2. Nem. Abakusszal is csak algoritmus szerint lehetett számolni al-Chvarizmi előtt is és utána is, sőt így marad ez a jövőben is. Az algoritmus-fogalom (nem az algoritmus név!) és az algoritmus-használat sokkal régebbi al-Chvarizminél, mint ahogyan ezt az ókori euklideszi algoritmus és más ókori abakusz-algoritmusok bizonyítják.

3. Attól függ, hogy hogyan értelmezzük a szögfüggvény szót. Ha olyan függvényt értünk rajta, amely szögeken van értelmezve, akkor nyilvánvaló, hogy nem minden szögeken értelmezett függvény periodikus, mint ahogyan például a $\sin(x) + \sin(\pi x)$ sem az. Ha a szögfüggvény olyan függvény, amelynek értéke szög, akkor sem lesz minden ilyen függvény periodikus. Példák lehetnek erre egy torziós szál elcsavarodási szöge a forgatónyomaték függvényében, vagy egy szakasz látószöge egy adott egyenes pontjaiból. Ha viszont a szögfüggvényeket úgy értelmezzük, hogy szögfüggvény csak és kizárólag az iskolában tanult $\sin(x)$, $\cos(x)$, $\mathrm{tg}(x)$, $\mathrm{ctg}(x)$, $\sec(x)$, $\mathrm{cosec}(x)$ függvény, akkor igaz lesz az, hogy „a szögfüggvények periodikusak". E definíció szerint azonban például $\sin(x) \cdot \cos(\pi x)$ függvényt nem szabad szögfüggvénynek, más néven trigonometrikus függvénynek tekinteni.

4. A törtvonal egyetlen egyenesszakasz, amely törtek (nem pedig törötteket!) számlálójának és nevezőjének elválasztására szolgál. A töröttvonal pedig szakaszláncnak, amely egymáshoz végpontjukkal csatlakozó egyenesszakaszokból áll. Az egyenesszakasz egyelemű szakaszlánc, egyelemű töröttvonal. Noha a törtvonal is egyenesszakasz, mégsem lenne szerencsés a törtvonalat speciális töröttvonalként felfogni, mert ez – sokéves tapasztalat szerint – logikai törést szokott okozni, nemcsak azért, mert a törtvonal aritmetikai, a töröttvonal pedig geometriai fogalom (ez még nem volna elegendő ok), hanem azért is, mert ettől törtelnül vezet az út odáig, hogy valaki a töröttvonalat többszörös törtvonalként értelmezze. A törtvonal törtszámok jelölésénél használt vonal, de nem ő van törve. A töröttvonalnak pedig semmi köze a törtszámokhoz, viszont ő maga törött, olyan értelemben, hogy törések lehetnek benne.

5. Közkedvelt rövidítés például a *program counter*, a *programmable controller* és a *personal computer* szavakra.
A *program counter* („programszámláló") az eredeti nyelvben sem szerencsés kifejezés, ugyanis nem programszámláló, amit jelöl, még csak nem is a programhoz tartozó számláló, ugyanis nem számláló, hanem inkább címmutató, hiszen legtöbbször az éppen végrehajtás alatti vagy a következő végrehajtandó utasítás címének regisztrálására szolgál. Egymásutáni címeken levő, és ilyen sorrendben végrehajtandó utasítások esetében a címmutatás számlálással történik ugyan, más vezérlésátadások esetében azonban nem. Az IP, azaz instruction pointer (= utasítás (cím) mutató) kifejezés sokkal sikerültebb, és kerdi is kiszorítani a PC-t ebből az értelemből. A *programmable controller* programozható, (általában digitális) szabályozó vagy vezérlő berendezés, vagy inkább annak központi része. A *personal computer* pedig az ún. személyi számológép, amelyből ma már vannak olyan példányok is, amelyek több munkahelyesek, több személy is használhatja őket egyszerre, és áruk olyan, hogy egy személynek, ha csak közönséges személy (és nem egyben milliomos személy is) nemigen van annyi pénze, hogy megvásárolhasson és személyi tulajdonának tekinthessen egy ilyen eszközt.

6. Az utóbbi.

7. Általánosan: olyan szám, amit ellenőrzésre használnak. Tudatos hamisítás és véletlen tévesztés észrevételét lehet megkönnyíteni úgy, hogy a szóban forgó számból (vagy számokból) valamilyen szabály szerint egy másik számot (más számokat) képezünk – ez(ek) az ellenőrző szám(ok) – és az eredetit (eredetiket) ezekkel egyesítjük (összekeverjük), és ezeket az így nyert egységeket bocsátjuk az információforgalomba. Lehet az ellenőrző szám például az eredeti szám 9-cel való osztási maradéka, amit a szám végéhez egyszerűen hozzáírunk, de használhatunk többjegyű is sokkal bonyolultabban kiszámított ellenőrző számokat is. Természetesen nemcsak egyes számok, hanem például számsorozatok, számtáblázatok, információállományok is „védhetők" ily módon.

8. Van. A negatív (például mínusz kettő) alapú számábrázolási rendszerben például az előjelnek nemcsak hogy fenntartott helye nincs, hanem egyáltalán nincs helye, ugyanis nincs szükség az előjel külön jelzésére.

9. Van. Például az 1/2 alapú helyiértékes számábrázolási rendszerben. Egy szám számjegyei ugyanis ebben a rendszerben megegyeznek ugyanennek a számnak a 2 alapú számábrázolási rendszerben felírt alakjának jegyeivel, csak sorrendjük ellentétes. Így az átvitel is ellentétes irányú lesz.

10. A mix a mixtura (keverék) szó rövidítése. A számítástechnikai mix leggyakrabban súlyozott összeg (a keverési arányokat a súlyokkal adják meg), melynek tagjai a számológép vagy valamely program jellemző értékeit (például elvégzési időket) tényezőként tartalmazó szorzatok.
A mixeket az átlagokhoz hasonlóan durva becslésekre, durva jellemzésekre használják.

11. Nem is egy. Néhány ilyen típus a következő: Epson PX–8, NEC, Olivetti M 10, Tandy Model 100, Apricot Portable, Data General One, Dulmont Magnum, HP 110, Osborne Vadem, Pro-Lite, Sharp PC 5000, Toshiba T 1100. A legnagyobb kijelzőfelülettel a Data General One rendelkezik.

12. 1. Egy irányításstechnikai műszertípus szabványos neve, melynek funkciója nyomáskijelzés és vészjelzés-adás. Nevét a pressure indication and alarm kezdőbetűiből alakították ki (pressure indication = nyomáskijelzés, alarm = riadó). 2. A peripheral interface adapter kezdőbetűiből alkotott szó. Szó szerinti fordítása a „perifériaillesztő egység", nem pontos. Teljesen egzakt a „perifériális egység illesztő egység" lenne.

13. A számvezérlésű szerszámgépek „magas szintű" programozási nyelvét két menetben fordítják. Az első fordítási menetet az ún. „processzor", a másodikat pedig az ún. „posztprocesszor" végzi. A processzor itt szoftver, egy fordítóprogram, nem azonos a számológép processzorával, ami hardver. A processzor az általános, minden konkrét számvezérlésű szerszámgépre egyformán értelmezett információkkal, a posztprocesszor (ez is fordítóprogram) pedig az adott konkrét számvezérlésű szerszámgépre vonatkozó információkkal foglalkozik, annak specialitásait figyelembe véve. A processzor szó szerinti jelentése eljárást végző. A posztprocesszor szóban a poszt jelentése „után", „utáni". A posztprocesszor egzakt jelentése „utánfeldolgozó", „utánfeldolgozást végző", vagy elnagyoltan „a processzor utáni".

14. b), c) és d) semmiképp sem. a) a nyelvileg pontos fordítás, de az eredeti nyelvben is legtöbbször hozzáértők még a lemezre írást és lemezről olvasást is. Ezért tartalmilag e) áll hozzá legközelebb. Az f) azért a legszerencsésebb, mert rövid és igaz ugyan, de sem az írásra, sem az olvasásra, sem pedig a mechanikus mozgatásra nem utal közvetlenül, de erre jelenleg nincs is szükség, mert sehol sem gyártanak lemezegységet, amely nem látná el mindhárom feladatot.

15. 1. Kiváló gondolkodó, matematikus és természettudós (Blaise Pascal, ejtsd Blez Pászkál (1623–1663)). 2. A nyomás SI egysége, N/m^2. 3. Nicklaus WIRTH által alkotott, 1968-ban közzétett kedvelt, széles körben elterjedt „magas szintű" programozási nyelv.

16. A computer aided design kezdőbetűiből alkotott szó. Jelentése: számológéppel segített (támogatott) tervezés.

17. A computer aided manufacturing kezdőbetűiből alkotott szó. Jelentése: számológéppel segített (támogatott) gyártás (termelés).

18. Csak az, hogy a komplex információ-deformáció interdiszciplinaritását sikerült a koncepciók és disszciplinák nem disztingvált diffúziójával a konfúzió nemzetközi élvonalának szintjére fokozni. És mindezt kizárólag hazai eszközökkel, importból származó differenciálás és integrálás nélkül, kizárólag informatikával. Az ilyen tudományos világrekordok, szellemi csúcsteljesítmények – figyeljük csak meg! – előbb-utóbb életszínvonalunkba is be fognak gyűrűzni, primér módon az olvasás révén, az olvasó életkedvére kifejtett hatásukkal, szekunder módon pedig akkor, amikor ez az élenjáró tudomány kezd majd termelőerővé válni népgazdaságunkban.

19. Az információátvitelnél és átmásolás között rendkívül fontos különbséget tenni. Az átvitelnél átvisznk valamit. Csak az a biztos, hogy ahová az átvitel történt, ott megjelenik, amit átvittünk. Az a megszokott, hogy az előző helyről (ahonnan az átvitel történt) addigra eltűnik, amit onnan átvittünk. Az átmásolásnál ez az eltűnés nem következik be.

20. Állapotjellemzésre szolgáló változókat szoktak flag (ejtsd: fleg), azaz zászló, vagy semaphore (ejtsd: szemafor), azaz szemafor névvel illetni, mert a zászló és a szemafor ősidők óta széles körben használt, tipikus indikátor, állapotjelző. A programokban a flag olyan jellemző (vagy rekesz), amelynek értéke (tartalma) a pillanatnyi állapotot jelzi. Ha a flag lehetséges értékeinek halmaza kételemű, „logikai változó"-nak is szokták hívni. Flag, illetve szemafor funkciót hardver és szoftver eszközökkel egyaránt meg lehet valósítani.

Nagy teljesítményű,
szép írásképű,
megbízható nyomtatót kíván használni
számítógépéhez?

Dolgozzon
a
Magyar Optikai Művek

ROMOM k6311

típ. mátrixnyomtatóval

Kapcsolható
a Commodore gépekhez
és minden olyan számítógéphez,
amelynek Centronix illesztője van

A kedvező árral
és szolgáltatási feltételekkel
rendelkező nyomtatót 1985-től
folyamatosan tudjuk szállítani vevőinknek

Gyártja:

Magyar Optikai Müvek
Budapest XII., Csörsz u. 35.
T.: 151-230

MOM BUDAPEST

Főbb jellemzői:

nyomtatási sebesség	**100 jel/s**
nyomtatási raszter	**9 × 7**
írásformák	**ferde írás, normál írás**
nemzetközi karakterkészlet	**szabadon betölthető**
nyomtatási pozíciók soronként	**80 jel/10 cpi**
	100 jel/12,5 cpi
	120 jel/15 cpi
sortávolság	**6 sor/inch**
papírtovábbítás	**súrlódóhengeres tüskés rendszerrel**
másolatok száma	**1 eredeti, 2 másolat**

Forgalmazza:

MIGÉRT

Műszer és
Irodagépértékesitö
Vállalat
Számitástechnikai és
Ügyvitelgépesitési Osztály
Budapest VI.,
Népköztársaság útja 2.
T.: 323-332

„MARS" MENETTELJESÍTMÉNY ADATRÖGZÍTŐ ÉS SZÁMÍTÓ RENDSZER

Finommechanikai és Elektronikus Műszergyártó Szövetkezet
BUDAPEST XI., KARINTHY FRIGYES ÚT 22.
1052 Pf. 55